AFRL-RH-WP-TR-2010-0133

THE U.S. AIR FORCE-DEVELOPED ADAPTATION OF THE MULTI-ATTRIBUTE TASK BATTERY FOR THE ASSESSMENT OF HUMAN OPERATOR WORKLOAD AND STRATEGIC BEHAVIOR

William D. Miller, Jr.

Consortium Research Fellow

JULY 2010
Interim Report

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING,
HUMAN EFFECTIVENESS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th Air Base Wing Public Affairs Office and is available to the general public, including foreign nationals.

Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC).

AFRL-RH-WP-TR-2010-0133 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//**signed//**
James Christensen
Program Manager
Collaborative Interfaces Branch

//signed//
William E. Russell
Chief, Collaborative Interfaces Branch
Warfighter Interface Division

//**signed//**
Michael A. Stropki
Warfighter Interface Division
Human Effectiveness Directorate
711 Human Performance Wing

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government"s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-07-10	Interim	01-10-2008 to 01-07-2010

4. TITLE AND SUBTITLE
THE U.S. AIR FORCE-DEVELOPED ADAPTATION OF THE MULTI-ATTRIBUTE TASK BATTERY FOR THE ASSESSMENT OF HUMAN OPERATOR WORKLOAD AND STRATEGIC BEHAVIOR

5a. CONTRACT NUMBER
FA8650-08-D-6801 TO 10

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62202F

6. AUTHOR(S)
William D. Miller, Jr.

5d. PROJECT NUMBER
7184

5e. TASK NUMBER
08

5f. WORK UNIT NUMBER
71840879

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
711 Human Performance Wing
Human Effectiveness Directorate
Warfighter Interface Division
Collaborative Interfaces Branch
Wright-Patterson AFB, OH 45433-7022

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
711 HPW/RHCP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RH-WP-TR-2010-0133

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
88 ABW Cleared 12/21/10; 88ABW-2010-6636.

14. ABSTRACT
The Multi-Attribute Task Battery (MATB) has been a keystone for research investigating workload, trust in automation, and other cognitive research since its inception in the early 1990's. To facilitate continued use in research today, the United States Air Force took steps to ensure that the task would continue to remain relevant, developing its own version, AF_MATB. This version preserves all of the features and appearance of the original, in addition to implementing new features aimed at aiding the researcher: AF_MATB includes utilities to customize task parameters and generate scripts in an automated fashion, as well as a detailed performance log at the end of every run.

15. SUBJECT TERMS
Workload, Strategic Behavior MATB, Multitasking, Task Battery, Simulator, Multi-Attribute Task Battery

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. James Christensen
Unclassified	Unclassified	Unclassified	SAR	152	19b. TELEPHONE NUMBER *(Include Area Code)* DSN: 785-8748

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK.

Table of Contents

Glossary xiii
Acknowledgements xiv
1.0 INTRODUCTION 1
2.0 SYSTEM REQUIREMENTS 2
3.0 AF_MATB Subtask Descriptions 3
3.1. System Monitoring 3
3.1.1. Gauges 4
3.1.2. Lights 7
3.2. Resource Management 9
3.3. Tracking 12
3.4. Communications 13
3.5. Scheduling 15
4.0 BEFORE USING AF_MATB 22
4.1. Installation 22
4.2. Task Response Commands 27
4.2.1. System Monitoring 27
4.2.2. Resource Management 28
4.2.3. Communications 30
4.3. System Commands 31
4.4. Event-Related Commands 39
4.4.1. Transition Logging 39
4.4.2. System Management 39
4.4.3. Resource Management 41
4.5. Task Options 43
4.5.1. Tracking 43
4.5.2. Resource Management 44
5.0 RUNNING AF_MATB 45
5.1. Performance Folder Structure 50
5.2. Performance File Descriptions 53
5.2.1. Master Event Log 53
5.2.2. Transition Log 53
5.2.3. Communication Log 54
5.2.4. Correct Gauge Responses 54
5.2.5. Correct Light Responses 54
5.2.6. Fuel Tank Values 54
5.2.7. Incorrect Communication Log 55
5.2.8. Incorrect Gauge Responses 55
5.2.9. Incorrect Light Responses 55
5.2.10. Tracking Coordinate Log 55
5.2.11. Performance Summary 55
6.0 AF_MATB PARAMETERS 58
7.0 SYSTEM REQUIREMENTS 59
8.0 PARAMETER CLASSES 60
8.1. Script-Critical Parameters 60
8.1.1. Resource Management Parameters 60
8.1.2. System Monitoring Parameters 61

8.1.3. Communications Parameters 62
8.1.4. Tracking Parameters 62
8.2. Task-Operating Parameters 63
8.2.1. Resource Management Parameters 63
8.2.2. System Monitoring Parameters 70
8.2.3. AF_MATB System Parameters 72
8.2.4. Communications Parameters 76
8.2.5. Tracking Parameters 82
9.0 UTILITY BUTTON DESCRIPTIONS 84
9.1. Default Value 84
9.2. Save Values 85
9.3. Set All Default Values 88
9.4. Load Parameters File 89
9.5. Clear All Entries 92
9.6. Save and Continue to Script Generator 93
9.7. Load and Continue to Script Generator 94
9.8. Save and Continue to AF_MATB 95
9.9. Load and Continue to AF_MATB 96
10.0 AF_MATB_SCRIPTGENERATOR "WHY DO I NEED THIS?" 97
11.0 SYSTEM REQUIREMENTS 98
12.0 4 KEY COMPONENTS 99
13.0 "HOW DOES THIS UTILITY WORK?" 100
14.0 SCRIPT GENERATOR OUTPUT 101
15.0 SCRIPT GENERATOR PARAMETERS 102
15.1. Sequence Length 102
15.2. Communications 103
15.3. Tracking 104
15.4. System Monitoring 106
15.5. Resource Management 107
16.0 CONDITION LIST 110
17.0 VARIABLES CURRENTLY LOADED 111
18.0 UTILITY BUTTON DESCRIPTIONS 112
18.1. Generate Script 112
18.2. Load Any Parameters File 116
18.3. Set All Default Values 118
18.4. Clear Script Parameters 119
18.5. Generate Script & Continue to MATB 120
18.6. Load Script & Continue to MATB 120
REFERENCES 122
APPENDICES 123
Appendix A: File Listing for AF_MATB System Files Folder 123
Appendix B: Example Performance File 124
Transition Log 124
Master Event Log 125
Communication Log 126
Correct Gauge Responses Log 127

Correct Light Responses Log 128
Fuel Values Log 129
Incorrect Communication Log 130
Incorrect Gauge Responses Log 131
Incorrect Light Responses Log 132
Tracking Coordinate Log 133
Performance Summary 134
Appendix C: Example Script File 135
Script Worksheet 135
Event Translation Key Worksheet 136

List of Figures

Figure 001 – The AF_MATB window once loading has been completed. 1
Figure 002 – The entire System Monitoring subtask AF_MATB. 3
Figure 003 – The group of gauges used in the System Monitoring subtask. 4
Figure 004 – The maximum and minimum points of normal operation for a gauge. 4
Figure 005 – The minimum and maximum points of the malfunctioning range for the lower and upper portion of the gauge. 5
Figure 006 – A gauge with a malfunction that was properly corrected. 6
Figure 007 – The Lights portion of the System Monitoring subtask with both lights shown to be functioning normally. 7
Figure 008 – The Lights portion of the Systems Monitoring subtask with both lights malfunctioning. 7
Figure 009 – The Resource Management and Pump Status windows of AF_MATB, with Pumps 1 through 6 on. 9
Figure 010 – The Resource Management and Pump Status windows illustrating a Pump 1 fault. 10
Figure 011 – The Resource Management and Pump Status windows illustrating one of the 2 autopilot modes. 11
Figure 012 – The Tracking subtask running in MANUAL mode. 12
Figure 013 – The Tracking subtask running in AUTOMATIC mode. 12
Figure 014 – The Communications subtask window with the default maximum and minimum frequencies shown for the NAV channels and default minimum and maximum frequencies for the COM channels. 13
Figure 015 – The Communications subtask where someone just locked in frequency 121.5 on channel COM1, as indexed by the green enter button. 14
Figure 016 – The default appearance of the Scheduling window. No events have been scheduled or scheduling has been disabled in this instance. 15
Figure 017 – A Scheduling window with a 5-minute trial set to a low Tracking difficulty, and 8 Communications events. 16
Figure 018 – A Scheduling window with a 5-minute trial set to a moderate Tracking difficulty, and 16 Communications events. 16
Figure 019 – A Scheduling window with a 5-minute trial set to a high Tracking difficulty, and 22 Communications events. 17
Figure 020 – A Scheduling window with a 5-minute trial set to one of the automatic Tracking modes, and 8 Communications events. 18
Figure 021 – A Scheduling window showing three 2-minute trials comprising one run. The Tracking difficulties are set to low, autopilot, and high, with 8, 5, 8 Communication events for three trials. 19
Figure 022 – A Scheduling window with two 5-minute trials, with the information regarding the first trial completely contained on the timeline, and the information regarding the other trial truncated to fit the timeline. 20
Figure 023 – The same trial as shown in Figure 022, but with some time elapsed to show how the information in the second trial grows to fill the timeline as time progress. 21
Figure 024 – The MCR Installer icon on a black desktop background. 22
Figure 025 – The Visual C++ Component Runtime installation window. 22
Figure 026 – The Mathworks Installer screen for the MCR. 23

Figure 027 – Click “Install” to install the MCR so that the 3 included .exes can be used. 23
Figure 028 – The zipped AF_MATB System Files folder. 24
Figure 029 – “Extract all” from the System Files folder. 25
Figure 030 – The unzipped AF_MATB System Files is what AF_MATB will need to properly execute. 25
Figure 031 – The three .exes and the unzipped AF_MATB System Files folder in a directory. You are now ready to run the task. 26
Figure 032 – The six buttons in the task window that the user can click on to perform actions for the System Monitoring subtask. 27
Figure 033 – The eight buttons in the task window that can be clicked on to perform actions for the Resource Management subtask. 30
Figure 034 – The five objects in the task window that can be selected to perform actions for the Communications subtask. 31
Figure 035 – The AF_MATB window after pressing *Space*. 32
Figure 036 – The AF_MATB window after pressing *Esc*. 33
Figure 037 – The space below the System Monitoring subtask, demonstrating what the window looks like when the timer and date are disabled. 34
Figure 038 – The AF_MATB window after *Esc* was pressed again, placing the task in the “ready” state. 34
Figure 039 – What the task looks like when paused. 35
Figure 040 – The Task window with script loading window after pressing *Home*. 36
Figure 041 – Failure to load a script. 37
Figure 042 – Script successfully loaded. 38
Figure 043 – The first dialog box you will see when executing AF_MATB, the Subject ID dialog. 45
Figure 044 – If parameters are loaded directly from AF_MATB_Parameters, you will see this dialog box. 45
Figure 045 – When a file has been successfully loaded, this message will be displayed. 46
Figure 046 – Dialog box asking if you would like to load custom task parameters or a script. 46
Figure 047 – Dialog box which asks which custom file you would like to load. 47
Figure 048 – Standard file selection utility in Windows which you can use to load custom parameters or scripts. 47
Figure 049 – Loading was not successful due to a corrupt script or parameters file. 48
Figure 050 – Standard directory selection utility in Windows used to identify the location of the AF_MATB System Files folder. 48
Figure 051 – The AF_MATB Task, ready to be used. 49
Figure 052 – Saving In Progress message, notifying the user to please wait until the data has finished saving. 49
Figure 053 – An example performance folder generated after an experimental run. 50
Figure 054 – The contents of the performance folder generated in Figure 053. 51
Figure 055 – The contents of one of the Trial folders illustrated in Figure 054. 52
Figure 056 – AF_MATB_Parameters window when launched. 58
Figure 057 – Pump Fault Duration. 60
Figure 058 – RMS Interval. 60
Figure 059 – RMS Target Value. 61

Figure 060 – Gauge Malfunction Timeout. 61
Figure 061 – Indicator Light Timeout. 61
Figure 062 – Target Communication Timeout. 62
Figure 063 – RMS Interval. 62
Figure 064 – Pumps 1 & 3 Flow Rates. 63
Figure 065 – Pumps 2 & 4 Flow Rates. 63
Figure 066 – Pumps 5 & 6 Flow Rates. 63
Figure 067 – Pumps 7 & 8 Flow Rates. 64
Figure 068 – Tank A & B Drop Rates. 64
Figure 069 – Tank A & B Maximum. 64
Figure 070 – Tank C & D Maximum. 65
Figure 071 – Tank A Starting Volume. 65
Figure 072 – Tank B Starting Volume. 65
Figure 073 – Tank C Starting Volume. 66
Figure 074 – Tank D Starting Volume. 66
Figure 075 – Seconds Before Tank Values are Updated. 66
Figure 076 – Autopilot Maximum Difference Between Tanks. 67
Figure 077 – Main Tank Autopilot Ranges. 67
Figure 078 – Supply Tank Autopilot Ranges. 68
Figure 079 – Hide Tank Changes in Autopilot. 69
Figure 080 – Gauge Speed. 70
Figure 081 – End of Range Delay Max (Cycles). 71
Figure 082 – Correct Fault Identification. 71
Figure 083 – Input Options. 72
Figure 084 – Enable Pausing? 72
Figure 085 – Allow Manual Event Triggering? 72
Figure 086 – Allow Subject to Manually Fix Pumps? 73
Figure 087 – Display Date and Time? 75
Figure 088 – Enable Scheduling? 75
Figure 089 – Caller ID. 76
Figure 090 – Comm Slot 1 Label. 76
Figure 091 – Comm Slot 2 Label. 77
Figure 092 – Comm Slot 3 Label. 77
Figure 093 – Comm Slot 4 Label. 78
Figure 094 – Slot 1 & 2 Freq. Ranges. 78
Figure 095 – Slot 1 & 2 Freq. Increments. 79
Figure 096 – Example of incongruent frequency limits and increment. 79
Figure 097 – Warning as a result of attempting to save incongruent frequency limit and increments parameters. 80
Figure 098 – Slot 3 &4 Freq. Ranges. 80
Figure 099 – Slot 3 & 4 Freq. Increments. 81
Figure 100 – Enter Confirmation Highlight (Cycles). 81
Figure 101 – Tracking Gain. 82
Figure 102 – Center Range. 82
Figure 103 – Autopilot Direction Limit. 82
Figure 104 – Manual Direction Limit. 83

Figure 105 – Button array and information section informing of a successful save. 85
Figure 106 – AF_MATB_Parameters window excerpt illustrating an attempt to save with an invalid parameter entry. 86
Figure 107 – AF_MATB_Parameters window excerpt illustrating an attempt to save with a missing parameter value. 86
Figure 108 – An example of an error log generated as a result of attempting to save a file with an invalid parameter.....87
Figure 109 – Standard Windows File Save dialog which the program uses to assist the user in saving parameters files.....87
Figure 110 – The AF_MATB_Parameters window after the Set All Default Values button was pressed. 88
Figure 111 – Button array and information section informing of a successful load.....89
Figure 112 – Standard Windows File Selection dialog which the program uses to assist the user in loading parameters files. 90
Figure 113 – Button array and information section informing of a failure to load. 91
Figure 114 – The AF_MATB_Parameters window after the Clear All Entries button was pressed.....92
Figure 115 – Button array and information section informing of the loading of AF_MATB_ScriptGenerator after a successful save.....93
Figure 116 – Button array and information section informing of the loading of AF_MATB_ScriptGenerator after a successful load. 94
Figure 117 – Button array and information section informing of the loading of AF_MATB after a successful save. 95
Figure 118 – Button array and information section informing of the loading of AF_MATB after a successful load. 96
Figure 119 – The Sequence Length section of the Script Generator‟s containing the default values. Condition column added for reference. 102
Figure 120 – The Communication Task section of the Script Generator‟s parameters containing default values. Condition column added for reference. 103
Figure 121 – The Tracking Task section of the Script Generator‟s parameters containing default values. Condition column added for reference. 105
Figure 122 – The System Monitoring Task section of the Script Generator‟s parameters containing default values. Condition column added for reference. 106
Figure 123 – The Resource Management section of the Script Generator‟s parameters containing default values. Condition column added for reference. 107
Figure 124 – Pump Shut-off notification if the number of Shut-offs exceed 50% of the number of Failures & Fixes. 108
Figure 125 – The Resource Management section shown with the Fuel Autopilot enabled for the Low Difficulty trials and the Moderate Difficulty trial being configured from Fuel Autopilot back to the standard Failures & Fixes and Shut-offs parameters. 109
Figure 126 – The Script Generator‟s List of Conditions section, which details how many trials and of what difficulties the trials in a particular script will be. 110
Figure 127 – The Variables Currently Loaded section, which details precisely what parameters the Script Generator will use when constructing and saving a script.....111

Figure 128 – AF_MATB_ScriptGenerator after someone has attempted to generate a script using invalid script parameters. 112
Figure 129 – An error log generated by AF_MATB_ScriptGenerator in response to the invalid parameters used in Figure 128 113
Figure 130 – The information section of the window, informing the user that script generation is in progress. 114
Figure 131 – Information section of the window, informing the user of a successful script generation. 114
Figure 132 – Information section of the window, informing the user of a failure to save due to no trials being entered into the Script Generator or another unforeseen generation error. 115
Figure 133 – The Excel Format? section, which allows the user to toggle between Excel 2003 and Excel 2007 formats for convenience. 115
Figure 134 – Information section, informing the user that both task parameters and script parameters were loaded from the selected file. 116
Figure 135 – Information section, informing the user that a task parameters file was selected, and as a result, only task parameters were loaded from that file. 117
Figure 136 – Information section, informing the user of a failure to load due to a corrupt script or other unforeseen loading error. 117
Figure 137 – The AF_MATB_ScriptGenerator window after the Set All Default Values button was pressed. 118
Figure 138 –The AF_MATB_ScriptGenerator window after the Clear Script Parameters button was pressed. 119
Figure 139 – When selecting Generate Script & Continue to MATB, the user will first be notified that the script is being constructed. Upon successful completion of script construction, the user will be informed that AF_MATB is loaded. Please see Figure 140 for an example of that notification. 120
Figure 140 – Upon successful loading of a script file, the user will be notified that the AF_MATB task is loading. 121

List of Tables

Table 1 – Breakdown of what keyboard commands are functional given the configuration of specific System Options for the input mode "*GUI Buttons Only."* 73
Table 2 – Breakdown of what keyboard commands are functional given the configuration of specific System Options for the input mode "*Keyboard Only"* or *"Both GUI and Keyboard."* 74

Glossary

711th HPW/RHCPA	711th Human Performance Wing, Collaborative Interfaces Branch, Adaptive Interfaces Section
AFRL	Air Force Research Laboratory
AF_MATB	Air Force Multi-Attribute Task Battery
DPI	Dots Per Inch
GUI	Graphical User Interface
MCR	MATLAB Compiler Runtime
NASA	National Aeronautics and Space Administration
RAM	Random Access Memory
RMS	Root Mean Square

Acknowledgements

I would like to thank Dr. Robert Ruskin and the Consortium Research Fellows Program for their unbridled support, as well as Maggie Funke and Kylie Bushroe for their invaluable contributions to the technical manual. Finally, I would like to thank Justin Estepp, Iris Davis, Dr. Scott Galster, and Dr. James Christensen for their time, guidance, and support during the development of the software and technical manual.

THIS PAGE INTENTIONALLY LEFT BLANK.

1.0 INTRODUCTION

The original MATB was developed by J. Raymond Comstock and Ruth J. Arnegard in 1992. Originally designed as a benchmark for a wide range of studies regarding operator performance and workload (Arnegard & Comstock, 1991), it has become a mainstay for psychological and psychophysiological research regarding cognitive workload, as well as the issue of trust in automation (Carmody & Gluckman, 1993; Parasuraman, Mouloua, & Hilburn, 1998), and psychophysiologically-controlled aiding (Wilson, Lambert, & Russell, 2000).

With changes in computer hardware over the last 18 years, the original MATB software has become all but unusable on modern systems. In order to facilitate compatibility on newer computer systems in government and educational laboratories, additional hardware and software support was necessary. Due to MATB‟s widespread use in research, the ability to customize the task to suit specific research areas was also added, allowing MATB to cater to an even larger research community. In conjunction with this, more detailed performance analyses were also added, as well as the ability to quickly create and customize experimental scripts to again suit the specific needs of researchers. The result of these additions is AF_MATB, which can be executed from any Windows PC. This new version of the original MATB preserves all of the design features of the previous version, while implementing a number of new features designed to provide maximum flexibility and customizability while improving ease-of-use for the researcher and participant. The new features do not impact comparisons with the previous software; exact replications of previous research are possible and facilitated by the new supporting tools.

Like the original version, AF_MATB consists of six windows that comprise the total battery. Each of the four subtasks is represented in a separate window (see Figure 001); these subtasks include: System Monitoring, Communications, Resource Management, and Tracking. The last two windows, which contain Scheduling and the Pump Status information, are resources that the user can utilize to improve performance during the task. Each of these windows will be discussed later in more detail.

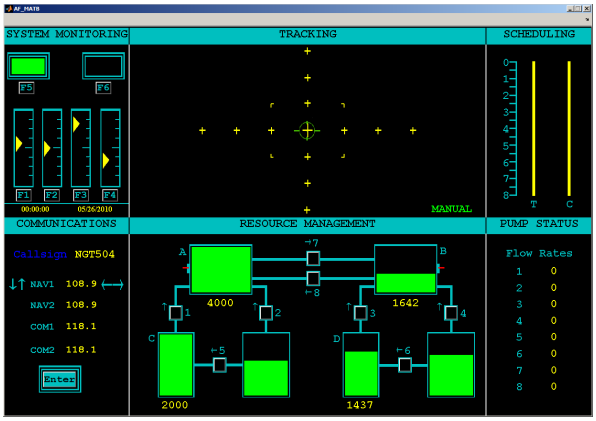


Figure 001 – The AF_MATB window once loading has been completed.

2.0 SYSTEM REQUIREMENTS

Hardware requirements for AF_MATB include a 1 GB of RAM, 2 GHz dual-core processor, a keyboard, mouse, joystick, and a 15-inch monitor (1280x1024 resolution), at minimum. It is important that these minimum requirements be met to ensure proper function of the task. Executing the task on machines that do not meet these minimum requirements may produce periods of freezing caused by audio playback, slower movement of the fuel level indicators, gauge indicators, and tracking crosshair, as well as unreliability of event times. Please note that while the task is sensitive to machines that do not meet the minimum requirements, movement in the task and playback of audio files will not be negatively affected in any way by machines that exceed these requirements.

A keyboard, mouse and joystick are all required for the task to be properly executed. A mouse is necessary for responding to some of the dialog boxes at the start of the task, and can be used by the subject to respond to events in the task. The keyboard is necessary for manual event triggering and AF_MATB System Commands, which will be discussed later. Finally, the joystick is required for operation of the Tracking Task. In the event that a joystick is not present, the user will be notified that no joystick has been detected at the start of the task and the tracking crosshair will float freely about the screen. Please be aware that the joystick must be connected to the computer prior to execution of the task. If the joystick is connected after the task has been executed, the joystick will not be detected.

AF_MATB was designed for Windows computers operating on Windows XP SP3 or higher. If the task is executed using a MATLAB script, then version 2007B or later should be used. Otherwise, MATLAB Compiler Runtime 7.8 or later is required.

Please ensure that your DPI settings are set to default values. If your DPI is set to custom values, you may experience window distortion.

3.0 AF_MATB Subtask Descriptions

3.1. System Monitoring

The System Monitoring subtask (see Figure 002) can be located in the upper left section of the AF_MATB window. It consists of two tasks: Gauges and Lights. The length of event faults can be manipulated using AF_MATB_Parameters, which will be discussed later in this manual. Please note that the performance and operating parameters for each of the two tasks are independent. For example, the Gauges and Lights components are each capable of having different timeouts.

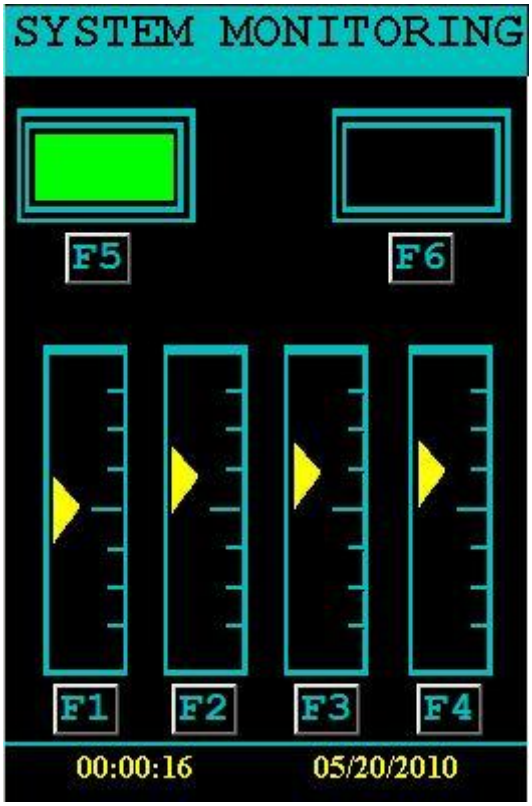


Figure 002 – The entire System Monitoring subtask AF_MATB.

3.1.1. Gauges

The Gauges subtask consists of the user observing the four gauges located in the System Monitoring window of AF_MATB (see Figure 003). The user must be informed of the normal operating behavior of the gauges, as well as behavior that would indicate a malfunction. The user is instructed that in event of a malfunction, a correct response is required within the experimenter-defined timeframe.

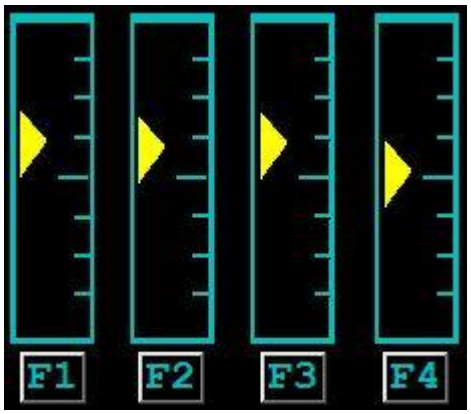


Figure 003 – The group of gauges used in the System Monitoring subtask.

Normal operating behavior for a gauge is defined as a gauge indicator smoothly fluctuating between one tick mark above the center of the gauge and one tick mark below the center of the gauge (see Figure 004). The gauge indicator may briefly pause at the top of its operating range before changing direction and moving towards the bottom of its operating range, or vice versa. The length of this pause is capable of being adjusted via the **End of Range Delay Max (Cycles)** parameter. Gauge speed can also be adjusted through the **Gauge Speed** *Lower* and *Higher* limit parameters/

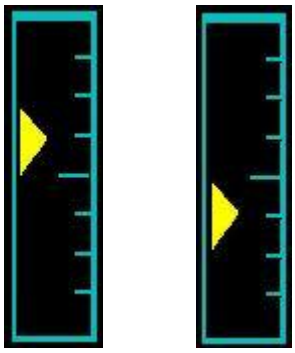


Figure 004a **Figure 004b**

Figure 004 – The maximum and minimum points of normal operation for a gauge.

The two images above illustrate the maximum and minimum value of each gauge when operating normally, respectively.

A gauge malfunction is defined as any time when the gauge indicator fluctuates outside of the previously defined operating range. This behavior will be exhibited for the duration specified by the **Gauge Malfunction Timeout**.

A gauge malfunction is characterized by a smooth transition out of the normal operating range and into either the upper or lower malfunctioning ranges. For the duration of the malfunction, the indicator will continuously alternate between the malfunctioning range and the normal range, as illustrated in Figure 005. The program is designed such that the scheduled malfunctions will occur in the area that the indicator is already traveling within. For instance, if the indicator is traveling up when the malfunction is scheduled to occur, the indicator will stay in the upper malfunctioning range. Conversely, if the indicator is traveling down when the malfunction is scheduled to occur, the indicator will stay in the lower malfunctioning range. The two sets of figures above (Figure 005) illustrate the minimum and maximum points of the malfunctioning range that can occur in the lower portion and upper portion of the gauge, respectively.

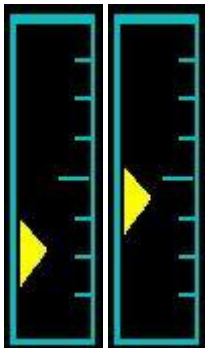


Figure 005a

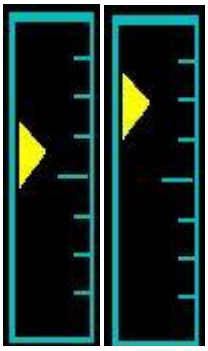


Figure 005b

Figure 005 – The minimum and maximum points of the indicator during a malfunction for the lower and upper portion of the gauge.

Responses for this task can be recorded in one of two ways. Users may respond using the keyboard command for each gauge (see **Keyboard Response Commands** for **System Monitoring**), or users may respond by clicking on the corresponding button below each gauge, i.e. ***F1***, ***F2***, etc. in the task window with the mouse. Please note that specific response modes may be enabled or disabled via **Input Options**.

If the appropriate response is recorded within the malfunction duration, the gauge indicator will automatically return to the center of the gauge, and stay there for a specified duration (**Correct Fault Identification)**. Additionally, a yellow bar will appear at the bottom of the gauge, as illustrated (Figure 006).

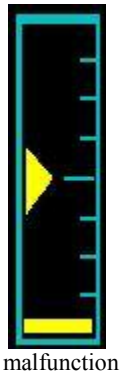


Figure 006 – A gauge with a malfunction that was properly corrected.

If no response is recorded within the malfunction duration, or if a response is recorded after the timeframe, it will be recorded as a timed-out response and a false alarm respectively. In the event that a response is not made in the correct timeframe, the gauge indicators will automatically resume normal functioning with no indication that a malfunction occurred.

3.1.2.Lights

The Lights subtask consists of the user observing two indicator lights located in the System Monitoring window of AF_MATB.

Figure 007 illustrates the normal behavior of the two lights. The light located on the left, designated Light 1, is green or "on" and the light located on the right, designated Light 2, is black or "off."

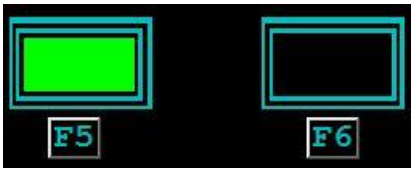


Figure 007 – The Lights portion of the System Monitoring subtask with both lights shown to be functioning normally.

In the event of a malfunction, each light exhibits different malfunctioning behavior, though the duration of the malfunction that either light exhibits will be the same. The duration of the malfunction can be manipulated via the **Gauge Malfunction Timeout** parameter.

Malfunctioning behavior that requires a user response for Light 1 is when the light "turns off" or turns black. For Light 2, malfunctioning behavior that requires a user response is when the light "turns on" or turns red. Each of these lights will malfunction for a set duration, as previously mentioned. The figure below (Figure 008) illustrates the malfunction for each light. Thus, if the user were to encounter the situation below, responses would be required for each light, as a malfunction has occurred for both Light 1 and Light 2.

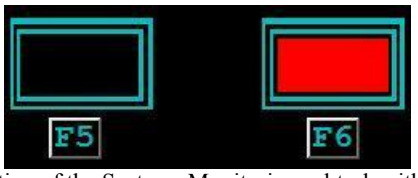


Figure 008 – The Lights portion of the Systems Monitoring subtask with both lights malfunctioning.

Responses for this task can be recorded in one of two ways. Users may respond using the keyboard command for each light (see **Keyboard Response Commands** for **System Monitoring**), or users may respond via the corresponding button below each light, i.e. ***F5*** or ***F6*** in the task window. Please note that specific response modes may be enabled or disabled via **Input Options**.

If the appropriate response is recorded within the malfunction duration, the malfunctioning behavior will automatically be corrected, and the light will be restored to its normal functioning behavior.

If no response is recorded within the malfunction duration, or if a response is recorded after the timeframe, it will be recorded as a timed-out response or a timed-out response and a false alarm, respectively. In the event that a response is not made in the correct timeframe, the lights will automatically resume their normal functioning behavior.

3.2. Resource Management

The Resource Management task is located in the bottom-middle section of the AF_MATB window. The goal of this task is to balance and maintain the two consumption tanks (Tank A and Tank B) at a specific volume using the eight pumps and four supply tanks. There are several parameters that can be manipulated through the Task Parameters Utility. These parameters include: the target volume, pump flow rates, tank starting volumes, and the maximum tank volumes. The amount of time that passes between updates of tank information can be manipulated through the Task Parameters Utility.

Please note that the window to the right of the Resource Management is the Pump Status Window (see Figure 009). This window is used to provide additional information regarding the pumps in the Resource Management task. It indicates which pumps are active by displaying the flow rate of each particular pump when it is on. In one of the autopilot options, all pump rates are replaced with "Auto," indicating that the fuel autopilot is active. The Pump Status Window is only meant to be used as a resource and requires no action on the part of the user.

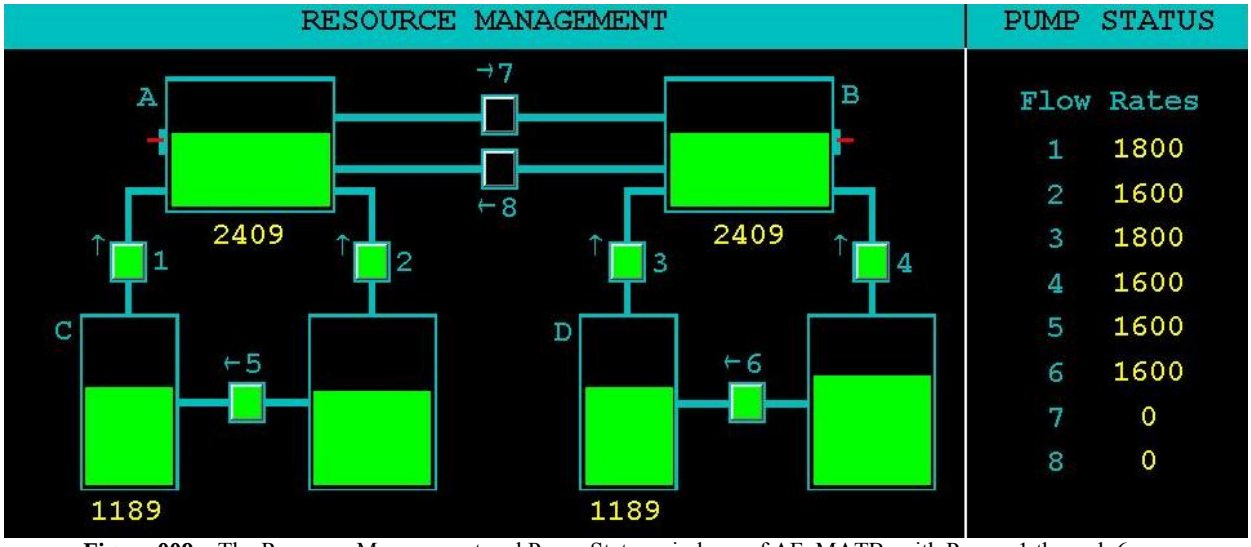


Figure 009 – The Resource Management and Pump Status windows of AF_MATB, with Pumps 1 through 6 on.

Tank A can be filled using three methods. The first way that fuel can be pumped into Tank A is through Tank C, via Pump 1. However, Tank C is not a bottomless tank, and must be filled from the bottomless supply tank to right of Tank C via Pump 5. The second way that fuel can be pumped in is directly from the bottomless tank via Pump 2. The third and final way fuel can be pumped into Tank A is from Tank B via Pump 8. This is a viable solution in the event that Tank B is over-filled while Tank A is under-filled or if Pumps 1, 2, or 5 are disabled for extended periods. However, it should be noted that this solution is not recommended except in specific cases due to the fact that, in most cases, it will still hurt one"s Resource Management score.

Tank B can also be filled using three methods. The first way that fuel can be pumped into Tank B is through Tank D, via Pump 3. However, Tank D is not a bottomless tank, and must be filled from the bottomless supply tank to the right of Tank C via Pump 6. The second way that fuel can be pumped in is directly from the bottomless tank via Pump 4. The third and final way fuel can be pumped into Tank B is from Tank A via Pump 7. As with transfers from B to A, this is a viable solution in the event that Tank A is over-filled while Tank B is under-filled or if Pumps 3, 4, or 6 are disabled for extended periods.

Pumps used in this task are toggled on and off, and this can be done in two ways. It can be done by pressing the corresponding number, i.e. ***1*** for Pump 1, etc. on the keyboard, or by mouse-clicking the small square in between two tanks. Squares that are black are considered "off" and are not pumping fuel, while boxes that are green are considered "on" and are pumping fuel.

The Resource Management subtask can experience two types of malfunctions or faults. The first type of fault is known as a Shut-off. In a Shut-off, a specific pump will turn off automatically with no notification that it is going to be turned off. In essence, a Shut-off simulates a user manually turning the pump off.

The second type of fault is a Pump Failure. In a Failure, the affected pump is turned off and locked (see Figure 010), such that the user will not be able to use that pump for a specified duration, which can be set via the **Pump Fault Duration** parameter. Please note that while it is not recommended for experimental conditions, the user can correct Pump Failures if the **Allow Subject to Manually Fix Pumps?** parameter is enabled.

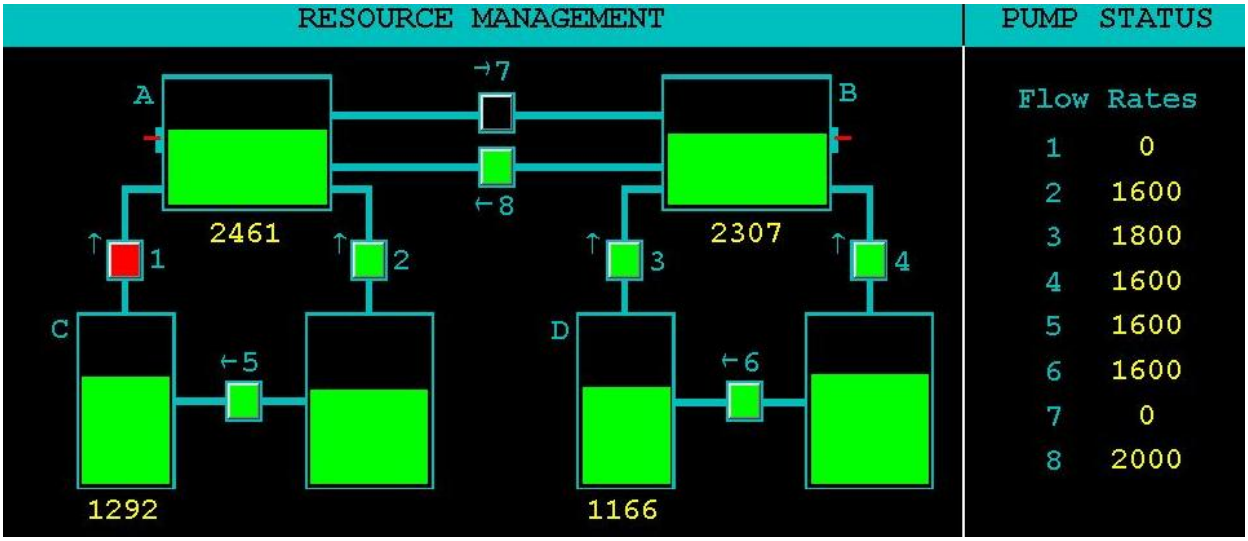


Figure 010 – The Resource Management and Pump Status windows illustrating a Pump 1 fault.

Pump Failures are indicated by the pump turning red for the specified duration of the event, as illustrated by Pump 1 in Figure 010. In addition, please note that within the Pump Status window, the flow rate for Pump 1 is 0, which indicates that no fuel is flowing through that pump.

At the end of the Failure duration, the pump will turn black, and normal function will be restored.

It should be noted that one of the new features of AF_MATB is an autopilot mode for the Resource Management subtask. This subtask is now one of two subtasks in AF_MATB that has an autopilot mode which can be enabled or disabled by either the script or the user. When the autopilot is engaged, the user‟s ability to manipulate pumps is disabled, and the computer will manage exactly which pumps should be on and off based on the **Autopilot Maximum Difference Between Tanks**, *Lower* and *Upper Limit* **Main Tank Autopilot Ranges**, and *Lower* and *Upper Limit* **Supply Tank Autopilot Ranges**. When a specific tank falls above or below one of these parameters, the autopilot will direct the correct pump to take the appropriate action.

There are two different modes that this autopilot can operate under. In the first mode, there is no indication that the Resource Management autopilot is on. There are no labels or notifications that it is active, except for the fact that pumps will change without any action from the user. Additionally, the user‟s actions are locked out while the autopilot is engaged.

The other mode, as illustrated in Figure 011, uses a less distracting approach. In this case, the autopilot hides all of its actions by turning all of the pumps blue and setting all Pump Status flow rates to “Auto.” This form of autopilot was created after users reported being distracted by the changes occurring in the Resource Management window while the autopilot was active. It should be noted that, using this method, the user‟s actions are still locked out when the autopilot is engaged.

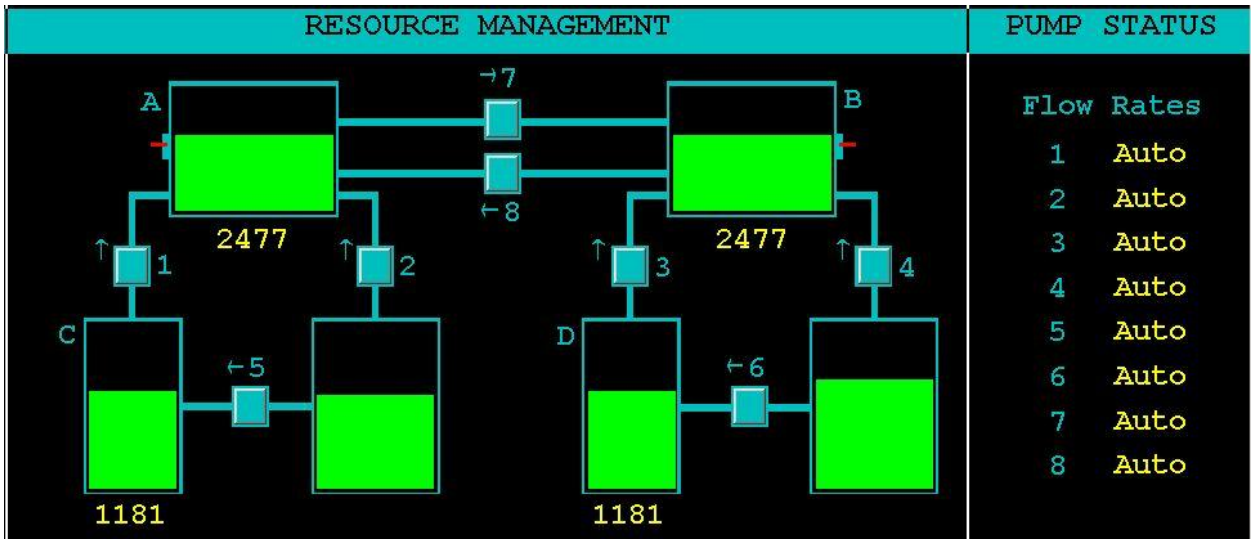


Figure 011 – The Resource Management and Pump Status windows illustrating one of the 2 autopilot modes.

Users can toggle between these two autopilot modes as long as the **Allow Manual Event Triggering?** parameter is enabled.

3.3. Tracking

Located in the middle upper portion of the Task Window, is the Tracking task. This subtask is the only subtask that does not have dual input. Using a joystick, the goal is to take the green crosshair and steer it as close to the center yellow crosshair as possible. Tracking has 3 difficulties, each of which can be triggered via the user or script. Each difficulty is marked by more frequent changes in direction and faster movement of the tracking crosshair.

This subtask is the only subtask, besides the Resource Management task, that includes an autopilot mode. When this autopilot mode is enabled, the "MANUAL" label (Figure 012) on the task will change to "AUTOMATIC" (Figure 013). Additionally, the tracking crosshair will move to the center crosshairs and stay there. During this time, no input from the joystick is required. This mode can be activated via the user or script, as well.

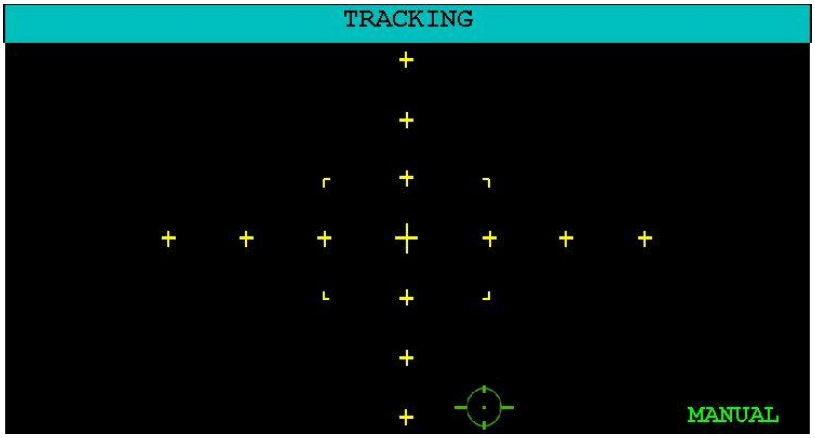


Figure 012 – The Tracking subtask running in MANUAL mode.

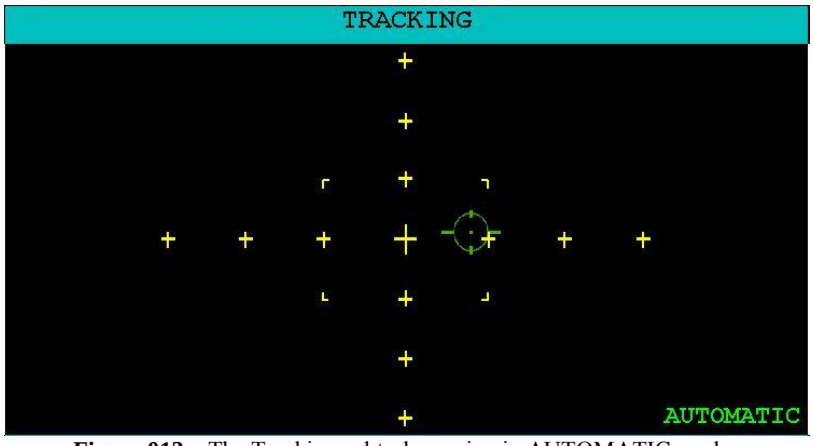


Figure 013 – The Tracking subtask running in AUTOMATIC mode.

Using the Task Parameter Utility, the speed of the crosshair, the frequency of direction changes (for both autopilot and manual modes), the center range for the autopilot, and the joystick gain may all be adjusted.

3.4. Communications

The Communications Subtask is an aural subtask located in the lower left portion of the Task Window. This task centers around 2 stimuli: True Communications Events and False Communication Events. True Events are addressed to the user‟s callsign designated by the blue “Callsign” label in the Communications Window (Figure 014). False Events are addressed to any callsign other than the one indicated in the Communications Window.

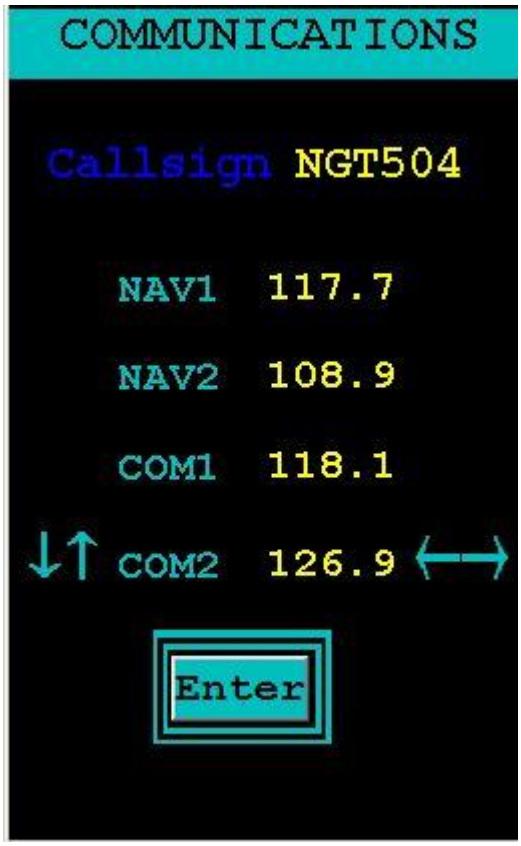


Figure 014 – The Communications subtask window with the default maximum and minimum frequencies shown for the NAV channels and default minimum and maximum frequencies for the COM channels.

True Events have a specified time-out associated with them. In order to successfully respond to a True Event, the user must follow the instructions issued and change the specified channel to the correct frequency. The four channels in Communications window are the column on the left, labeled as “NAV1,” “NAV2,” “COM1,” and “COM2” by default. The frequencies of each of those corresponding channels are located to the right of each channel.

An example of a True Event is as follows: “NGT504, NGT504, set first navigation one zero nine point seven.” “First navigation” refers to the channel, NAV1, and “one zero nine point seven” refers to the frequency, 109.7. In order to properly respond, the arrow indicators need to be located on the appropriate channel, in this case, NAV1. In order to do this, the user can use the corresponding arrow keys on the keyboard, or the user can click on the corresponding arrow in the Task Window. Next, the correct frequency should be selected using either the keyboard or Task Window arrows. Finally, while the arrows are still on the appropriate channel, the user needs to lock in that frequency using either the ***Enter*** key on the keyboard or the “Enter” button on the Task Window. The “Enter” button will briefly change from blue to green once it has been pressed; this allows the user to verify that his/her response was locked-in (Figure 015). Please note that whatever channel and frequency the arrows are on when ***Enter*** is pressed is the information that will be locked in.

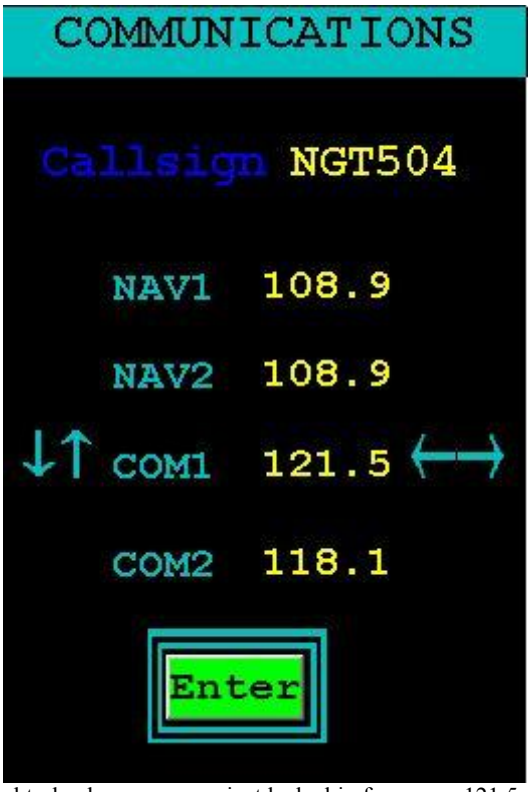


Figure 015 – The Communications subtask where someone just locked in frequency 121.5 on channel COM1, as indexed by the green enter button.

In the example above, the user has successfully locked in a frequency of 121.5 for the Com1 Channel. Please note, as previously stated, you can only lock in frequencies on the channel that the arrows were on at the time ***Enter*** or “Enter” was pressed.

Parameters that can be manipulated via AF_MATB_Parameters for this subtask include the channel labels, the True Event timeout, the channel frequency ranges, and the duration of the locked-in indicator.

3.5. Scheduling

The Scheduling Window is located in the upper right portion of the Task Window. This window allows the user to see eight minutes into the future for the Tracking and Communications subtasks. When scheduling is disabled, or when a script is not loaded, the scheduling window will appear with two thin yellow lines, as illustrated in Figure 016.

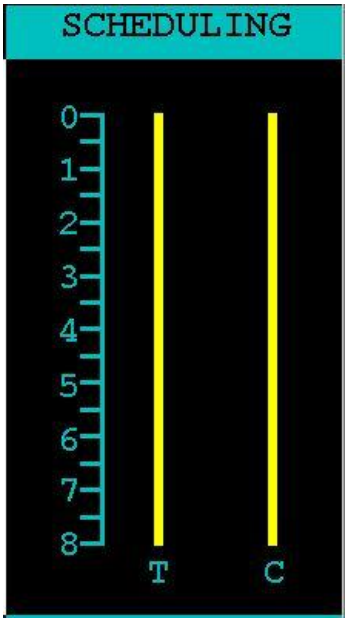


Figure 016 – The default appearance of the Scheduling window. No events have been scheduled or scheduling has been disabled in this instance.

For the Tracking subtask, the color and thickness of the line on the left, the "T" line, indicates different levels of tracking difficulty. For the Communications subtask, every communication event, regardless of whether it is a True Comm, or a False Comm, is signified by a small red rectangle on the right yellow line, or "C" timeline.

The use of this window may or may not be desired for some experimental paradigms, and as a result, it can be enabled or disabled through **Enable Scheduling?** parameter. This window requires no action on the part of the user (see Figure 017).

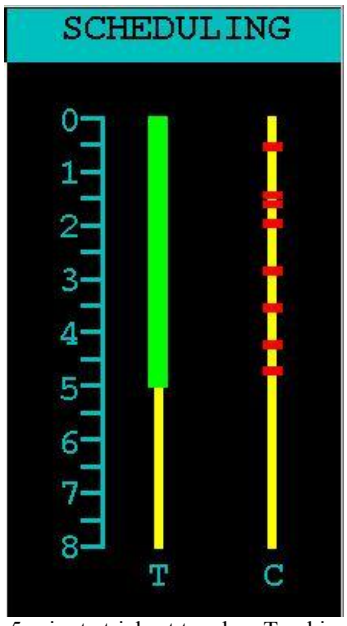


Figure 017 – A Scheduling window with a 5-minute trial set to a low Tracking difficulty, and 8 Communications events.

In this Scheduling window (Figure 017), the thicker green section on the "T" line, indicates that for the next 5 minutes, the Tracking Difficulty is set to "Low."Additionally, note the red rectangles on the "C" line. They indicate that in the next five minutes, eight communications will occur. As time progresses, the green bar will get shorter, indicating the progression of time, and the red rectangles will move closer to the top. Once they reach the top, they will disappear. A Scheduling window with a thicker yellow section indicates a moderate Tracking Difficulty (Figure 018). Also, note the increase in red rectangles on the Communications timeline.

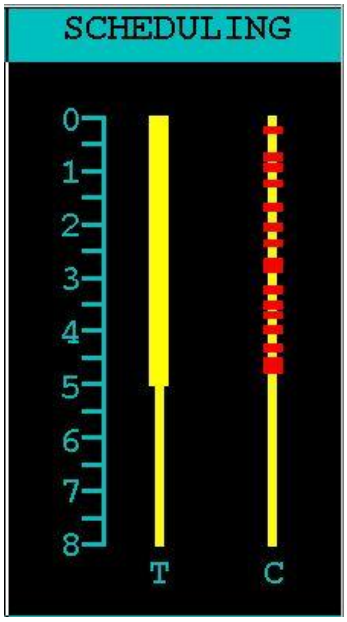


Figure 018 – A Scheduling window with a 5-minute trial set to a moderate Tracking difficulty, and 16 Communications events.

A Scheduling window with a thick red line indicates a high Tracking Difficulty (Figure 019). In addition, note the red rectangles on the communications timeline; with this schedule, the communications would be received almost one-after-another in many cases.

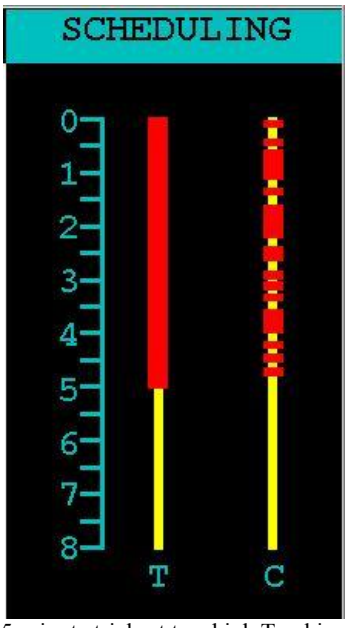


Figure 019 – A Scheduling window with a 5-minute trial set to a high Tracking difficulty, and 22 Communications events.

A Scheduling window with a thin blue line indicates that the Tracking Autopilot has been enabled for this trial (Figure 020). Please note that if the autopilot is manually controlled by the subject, those changes will not be reflected in the scheduling window.

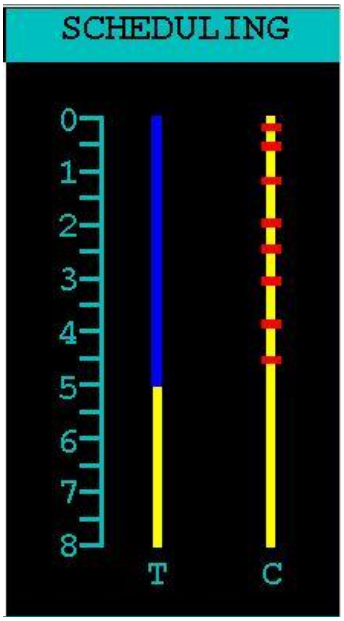


Figure 020 – A Scheduling window with a 5-minute trial set to one of the automatic Tracking modes, and 8 Communications events.

The Scheduling window is able to handle all trial conditions input via a script. In this instance (Figure 021), a six minute script was loaded, with three trials of two minutes each. In the first two minutes, the Tracking Difficulty was set to "Low," while the second 2 minutes, minutes 2:00 – 4:00 were set to autopilot, and finally, minutes 4:00 – 6:00 were set to "High."

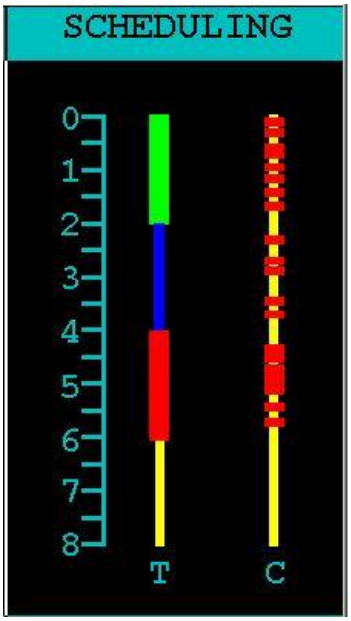


Figure 021 – A Scheduling window showing three 2-minute trials comprising one run. The Tracking difficulties are set to low, autopilot, and high, with 8, 5, 8 Communication events for three trials.

The Scheduling window is also able to cope with trials or scripts longer than the window. In this example (Figure 022), a 10 minute script was loaded comprised of two 5 minute trials.

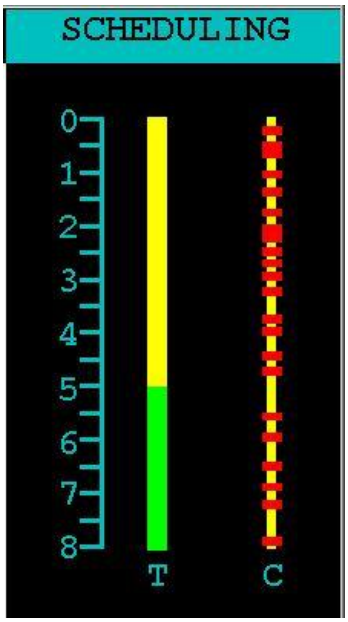


Figure 022 – A Scheduling window with two 5-minute trials, with the information regarding the first trial completely contained on the timeline, and the information regarding the other trial truncated to fit the timeline.

By comparing this figure (Figure 023) with the previous figure (Figure 022), one can see that as time progresses, the bar that indicates one‟s current tracking difficulty will shrink, while the next

visible bar will either grow to indicate its full duration (the green bar now indicates that the next trial is 5 minutes long, as opposed to the previous figure, when it could only indicate up to 3 minutes). Once the last bar on the timeline grows to indicate its true duration, it will start to move up the timeline as the bar before it continues to shrink.

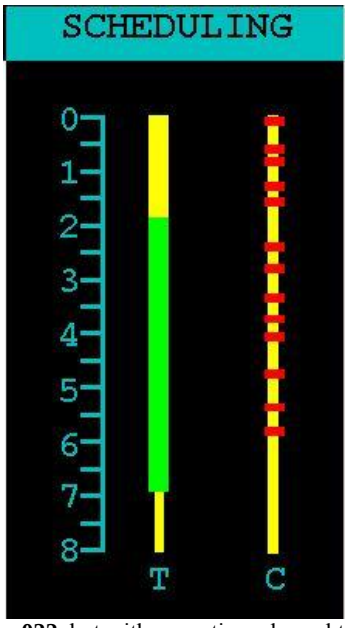


Figure 023 – The same trial as shown in **Figure 022**, but with some time elapsed to show how the information in the second trial grows to fill the timeline as time progress.

4.0 BEFORE USING AF_MATB

4.1. Installation

In order to run AF_MATB using the executable, an acceptable MATLAB Compiler Runtime (MCR) needs to be installed (see Figure 024). This version of the installer included in the AF_MATB package is MCR7.8, though any more recent version should work, as it has been verified to work with versions as recent as MCR 7.13.

In order to install the MCR, run the included InstallShield Wizard (see Figure 025).

Figure 024 – The MCR Installer icon on a black desktop background.

The initial wizard will then ask to choose your language, and then install the Visual C++ Runtime Components, called VCREDIST_X86.exe.

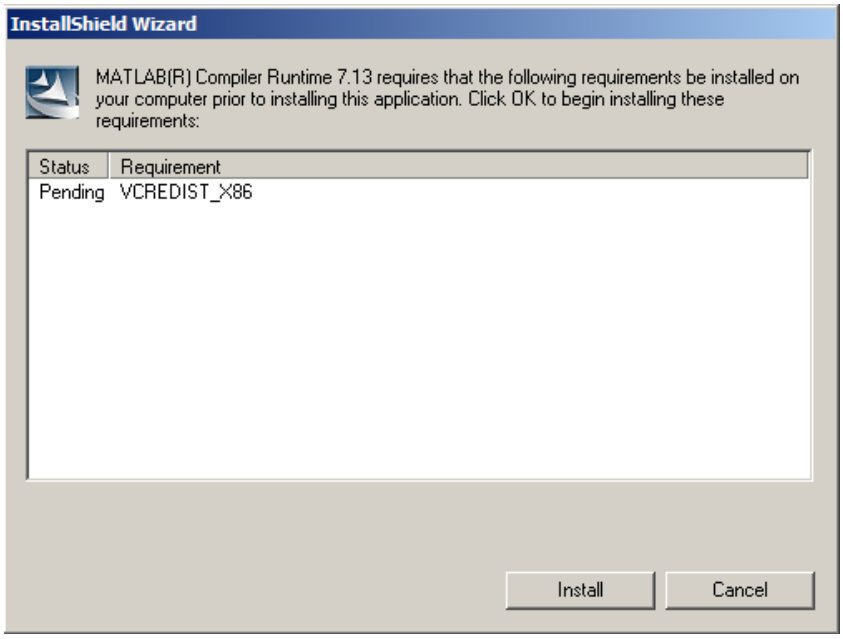


Figure 025 – The Visual C++ Component Runtime installation window.

These runtime components will extract and install. After this, the Mathworks Installer screen (see Figure 026) will appear.

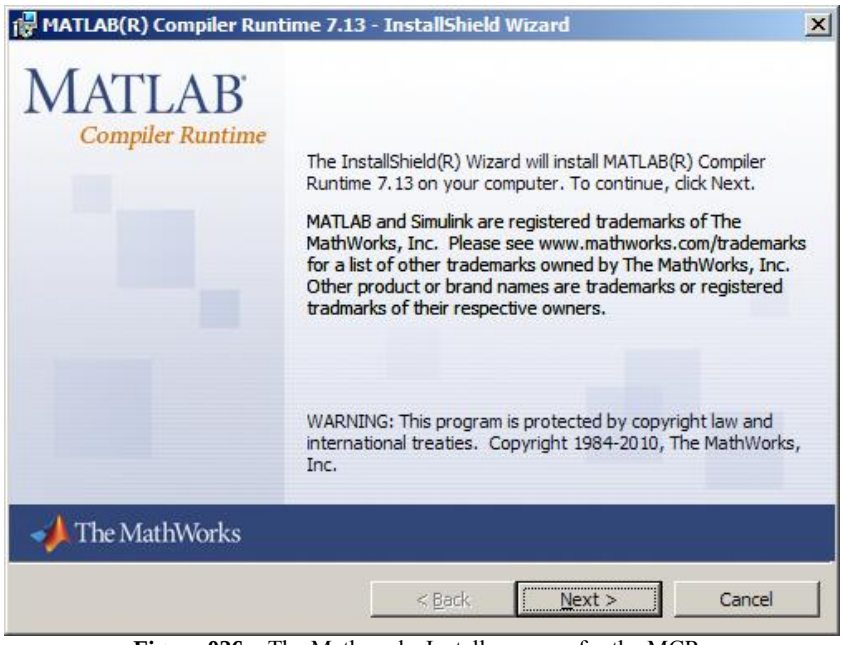


Figure 026 – The Mathworks Installer screen for the MCR.

Click "Next" and select your Name and Organization, as well as where you are installing the files, and then click "Install" (see Figure 027).

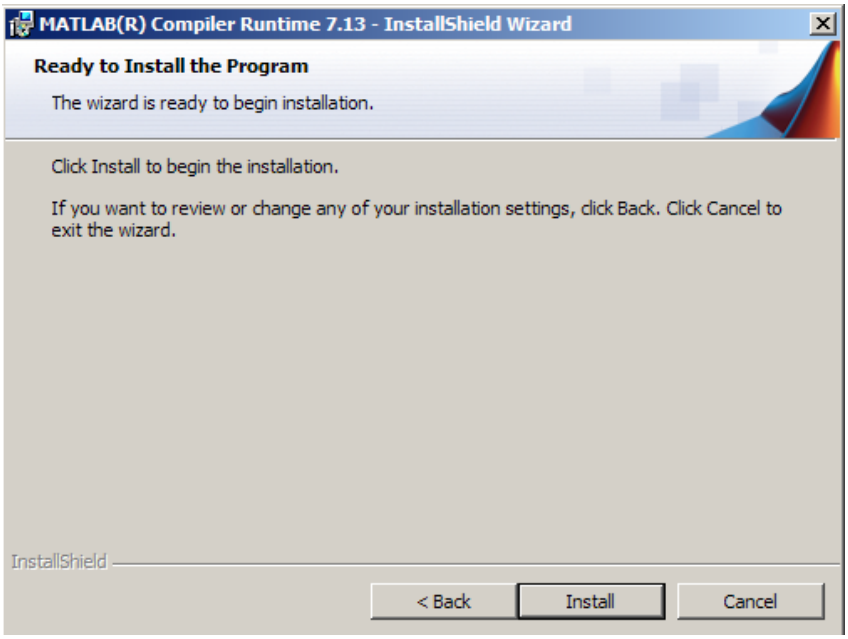


Figure 027 – Click "Install" to install the MCR so that the 3 included .exe files can be used.

Once the installation has completed, you are able to run all three of the included .exe files. After installing the MCR, the System Files used by the task need to be unpacked. Simply right click on the icon and select “Extract All” (see Figure 028).

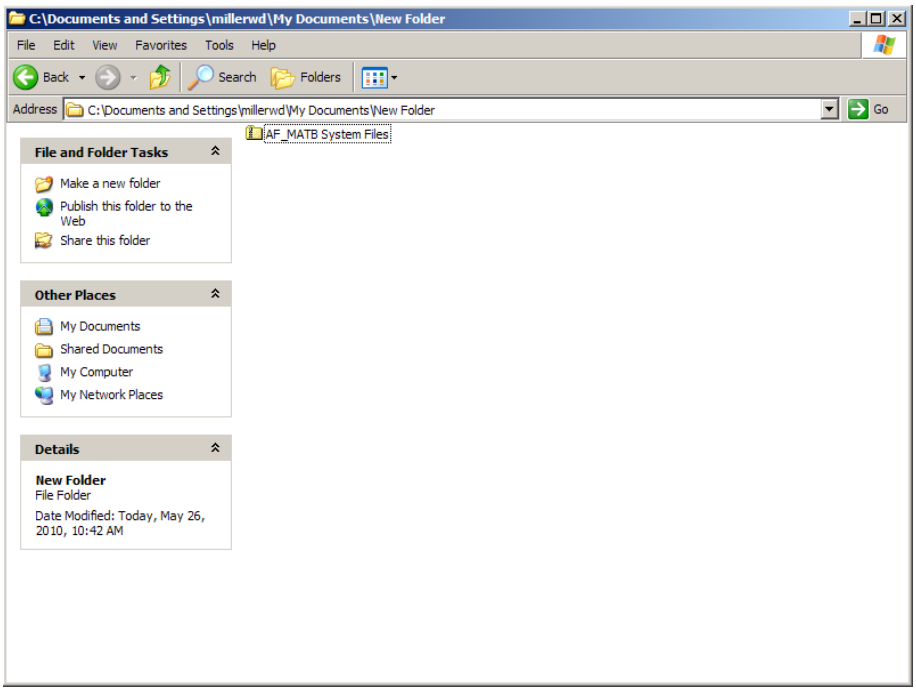


Figure 028 – The zipped AF_MATB System Files folder.

After extracting the files, an unzipped folder will appear with the same name (see Figure 029). This folder is the AF_MATB System Files folder utilized by the task. At this point, you can delete the .zip file.

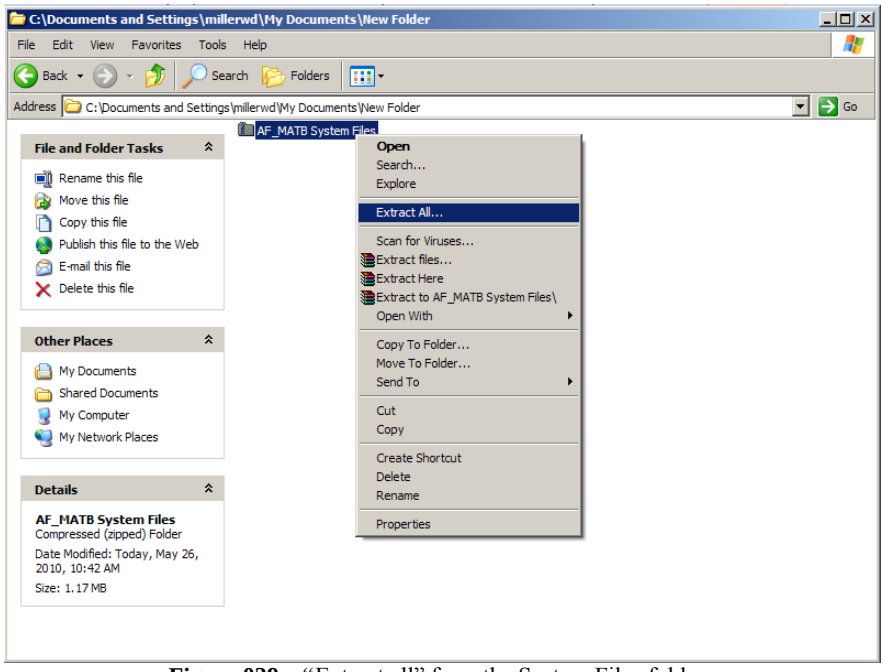


Figure 029 – "Extract all" from the System Files folder.

This folder is what AF_MATB will ask for after inputting the Subject ID and script information (see Figure 030). All that is left is to copy the three .exe files into the directory that the System Files folder is located in and execute (see Figure 031). The System Files folder should not be modified, nor should anything be

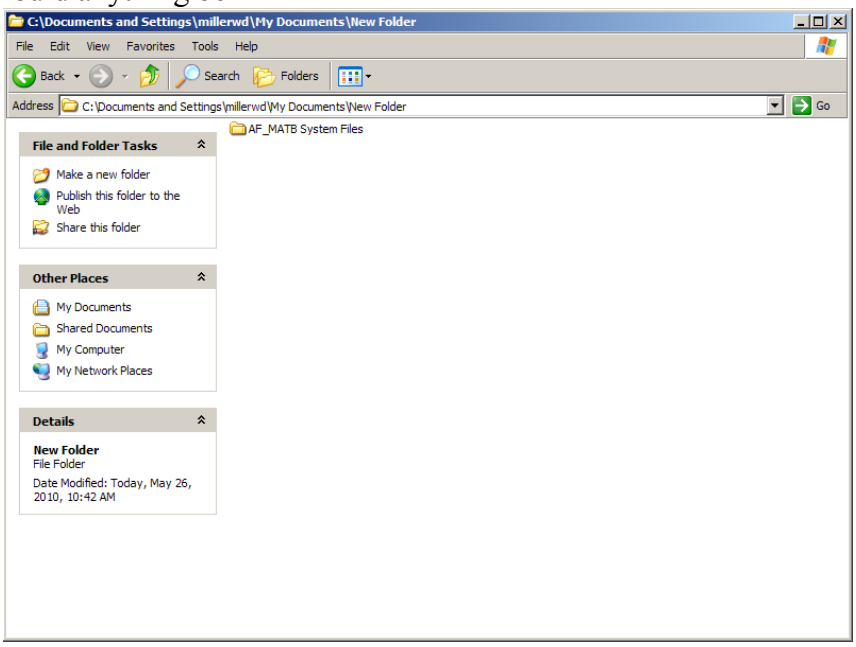


Figure 030 – The unzipped AF_MATB System Files is what AF_MATB will need to properly execute.

placed in this folder. If any additional files are added to the System Files folder, the task will not execute.

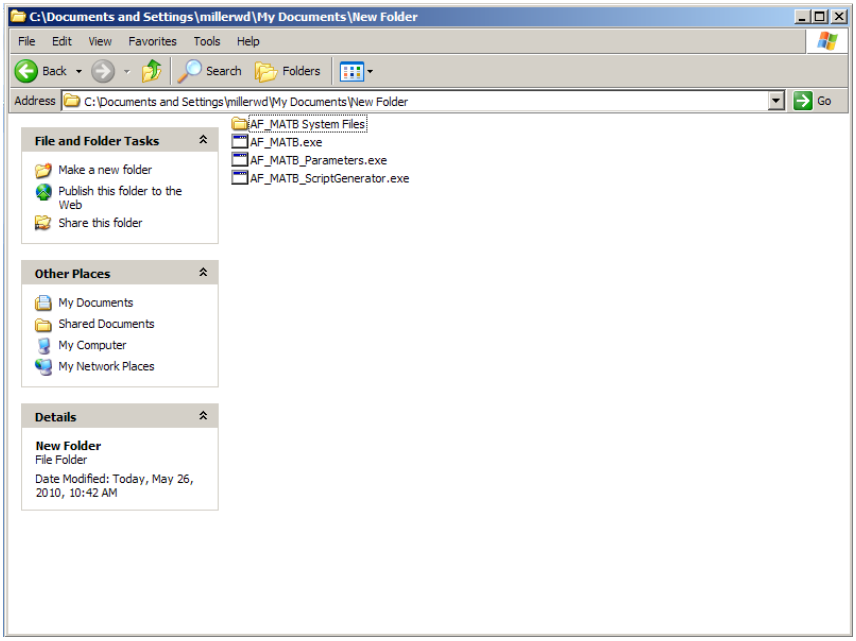


Figure 031 – The three .exes and the unzipped AF_MATB System Files folder in a directory. You are now ready to run the task.

4.2. Task Response Commands

4.2.1. System Monitoring

F1

In the event of an error in the first gauge, pressing this key (see Figure 032a) will fix the first gauge. If there is no error in the first gauge, then pressing this key will log a "False Alarm," which is the case for all gauge and light keyboard commands in the **System Monitoring** section. Please note that the figures (Figure 032) are the buttons in the task.

Figure 032a

F2

In the event of an error in the second gauge, pressing this key will fix the second gauge (see Figure 032b).

Figure 032b

F3

In the event of an error in the third gauge, pressing this key will fix the third gauge (see Figure 032c).

Figure 032c

F4

In the event of an error in the fourth gauge, pressing this key will fix the fourth gauge (see Figure 032d).

Figure 032d

F5

In the event of an error in the first light, which can be either green or black, pressing this key will fix the first light (see Figure 032e).

Figure 032e

F6

In the event of an error in the second light, which can be either red or black, pressing this key will fix the second light (see Figure 032f).

Figure 032f

Figure 032 – The six buttons in the task window that the user can click on to perform actions for the System Monitoring subtask.

Users can click on a button in the task window using the mouse, instead of using the function keys on the keyboard to have it perform the same function as you would by pressing the corresponding keyboard key.

4.2.2. Resource Management

1

The first pump can be toggled on or off using either the corresponding keyboard command (in this example, the number ***1*** key) or by clicking on that pump with the mouse (see Figure 033a). The direction of flow from this pump, as denoted by the arrow next to that pump, indicates that fuel is flowing from Tank C to Tank A at the rate indicated by the Pump Status Window.

Please be aware that all pumps in the **Resource Management** subtask are capable of being toggled on or off using either the keyboard command, or by clicking on the pump with the mouse. In addition, no pump will be able to be turned on if fuel is unable to flow from it. Finally, any pump that is experiencing a fault, as indicated by being red as opposed to black or green, will not pump fuel and will be stuck in the "Off" position for the specified fault duration. These three pieces of information are true for all pumps in the **Resource Management** subtask. Please note that the items that comprise Figure 033 are buttons in the task.

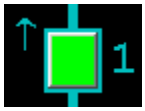


Figure 033a

2

The direction of flow from this pump, as denoted by the arrow next to that that pump, indicates that the fuel is flowing from an unlabeled supply tank to Tank A at the rate indicated by the Pump Status Window (see Figure 033b).

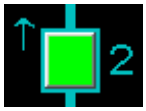


Figure 033b

3

The direction of flow from this pump, as denoted by the arrow next to that pump, indicates that the fuel is flowing from Tank D to Tank B at the rate indicated by the Pump Status Window (see Figure 033c).

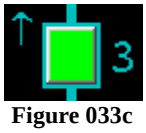


Figure 033c

4

The direction of flow from this pump, as denoted by the arrow next to that pump, indicates that the fuel is flowing from an unlabeled supply tank to Tank B at the rate indicated by the Pump Status Window (see Figure 033d).

Figure 033d

5

The direction of flow from this pump, as denoted by the arrow on top of the pump, indicates that the fuel is flowing from an unlabeled supply tank to Tank C at the rate indicated by the Pump Status Window (see Figure 033e).

Figure 033e

6

The direction of flow from this pump, as denoted by the arrow on top of the pump, indicates that the fuel is flowing from an unlabeled tank to Tank D at the rate indicated by the Pump Status Window (see Figure 033f).

Figure 033f

7

The direction of flow from this pump, as denoted by the arrow on top of the pump, indicates that the fuel is flowing from Tank A to Tank C at the rate indicated by the Pump Status Window (see Figure 033g).

Figure 033g

8
The direction of flow from this pump, as denoted by the arrow on top of the pump, indicates that the fuel is flowing from Tank B to Tank A at the rate indicated by the Pump Status Window (see Figure 033h).

Figure 033h

Figure 033 – The eight buttons in the task window that can be clicked on to perform actions for the Resource Management subtask.

Users can click on the pump icon in the task window (the green squares in the examples) using the mouse, instead of the keys on the keyboard.

4.2.3. Communications

Please note that the five items that comprise Figure 034 under each of the five keyboard commands for the Communications Subtask correspond with that keyboard command. The first four of these are not buttons, however, but clickable images. Click on the arrow in the task window, to perform the corresponding action.

→ (Right Arrow Key)
Will increase the frequency of the channel that the arrows are currently on by the increment defined in the Task Parameters Utility. If the maximum frequency for that channel is reached, then the frequency will wrap around and start over from the minimum frequency (see Figure 034a).

Figure 034a

← (Left Arrow Key)
Will decrease the frequency of the channel that the arrows are currently on by the increment defined in the Task Parameters Utility. If the minimum frequency for that channel is reached, then the frequency will wrap around and start over from the maximum frequency (see Figure 034b).

Figure 034b

↑ (Up Arrow Key)
Allow the experimenter to select the channel they would like to manipulate. Pressing this key will move the arrows towards the top communication channel. If the first communication channel is selected and this arrow is pressed, the arrows will wrap around to the bottom and the fourth channel will then be selected (see Figure 034c).

Figure 034c

↓ (Down Arrow Key)
Allow the experimenter to select the channel they would like to manipulate. Pressing this key will move the arrows towards the bottom communication channel. If the bottom communication channel is selected and this arrow is pressed, the arrows will wrap around to the top and the first channel will then be selected (see Figure 034d).

Figure 034d

Enter
Locks in the frequency of the currently selected channel. Whichever channel is currently selected, as indicated by the location of the arrows, will be the channel and frequency that are recorded. The *Enter* button in the task will turn green for a short time (time configurable using the Task Parameters Utility), confirming that the frequency has been locked-in (see Figure 034e).

Figure 034e

Figure 034 – The five objects in the task window that can be selected to perform actions for the Communications subtask.

4.3. System Commands

Space
Designed to start the experiment. When a trial or practice run is ready to be started, pressing the space bar will trigger a countdown to start the task. This key will only respond if the task is in a "ready" state. The below figure (Figure 035) illustrates the countdown timer at the start of the program.

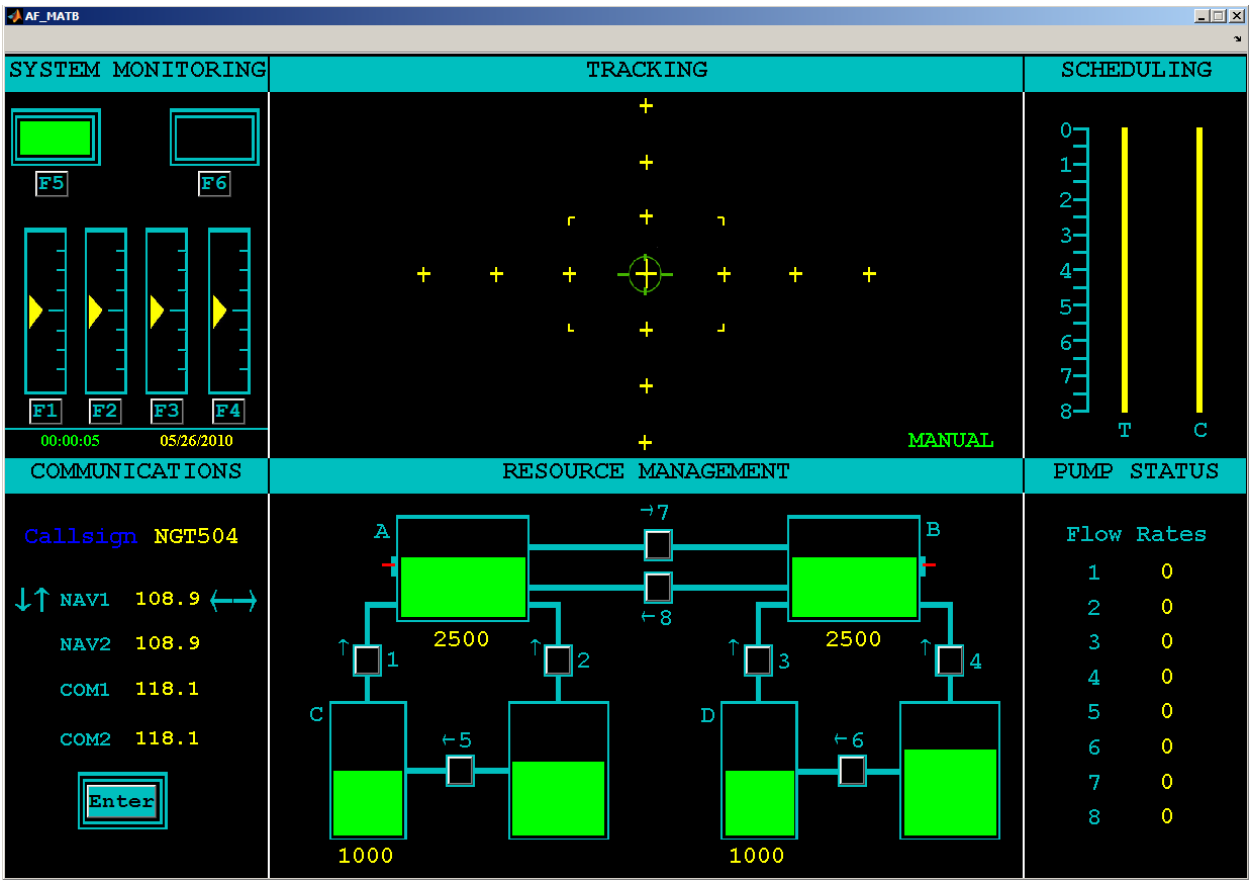


Figure 035 – The AF_MATB window after pressing ***Space***.

After pressing start, a five-second timer will appear in the upper left portion of the task window, below the gauges, regardless of whether **Display Date and Time?** is enabled. The timer is located at the bottom of the System Monitoring window. When the timer expires, the task will begin normal functioning. This countdown timer was added for experimental purposes to allow the system timer to ramp up and for data collection to be better coordinated.

Esc (Escape)
Designed to stop or restart the experiment. Pressing the escape key once will stop the trial or practice. Once stopped, the current trial/experiment cannot be resumed. At this point, the task is in a "suspended" state, as indexed by the red "Stopped." (Please note that if the date and timer have been disabled via the **Display Date and Time?** parameter, then "Stopped" will not appear.)

Figure 036 shows an example of a practice that has been stopped and is now in the "suspended" state. However, if the date and time are disabled, "Stopped" or "Paused" will not appear below the System Monitoring window when the task is stopped or paused. Figure 037 illustrates the absence of the date and timer, as well as other indicators like "Paused" or "Stopped".

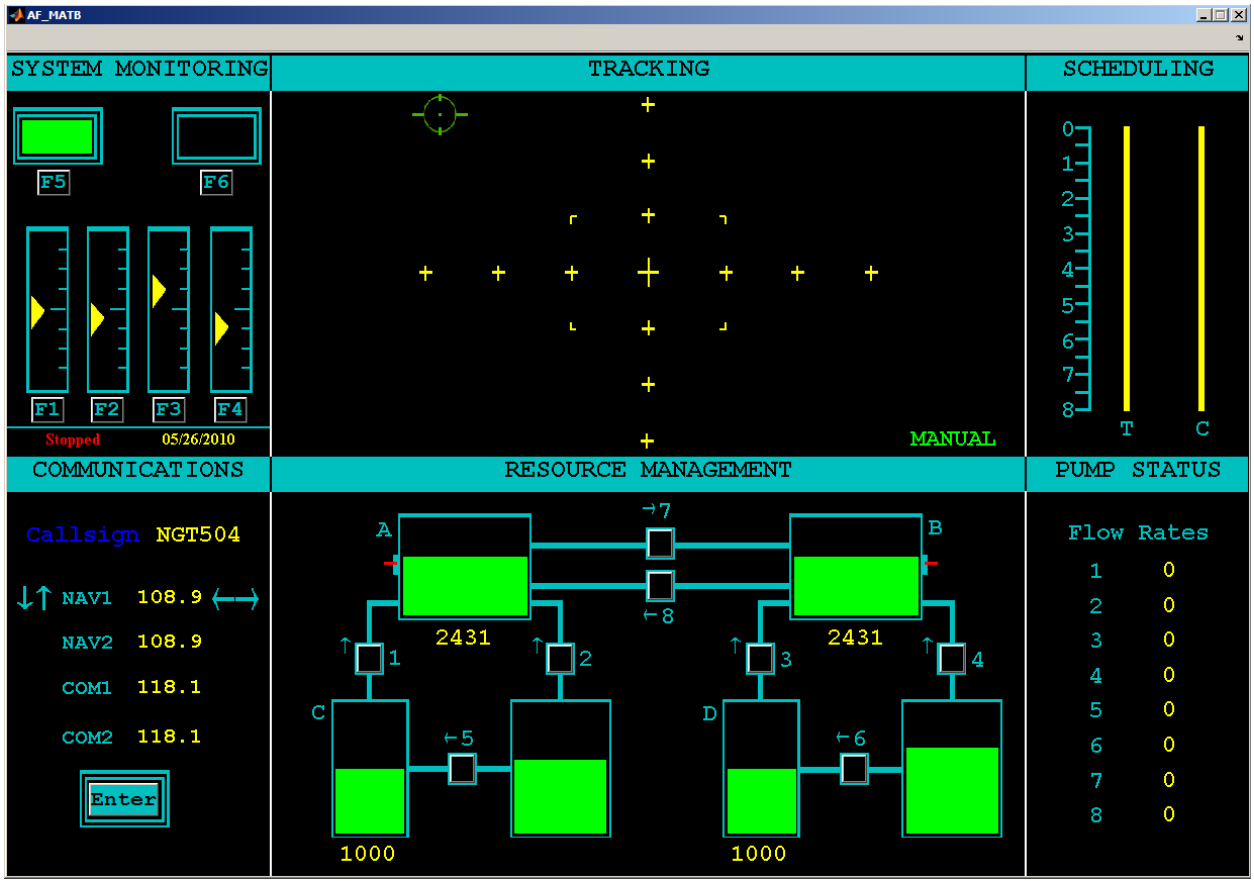


Figure 036 – The AF_MATB window after pressing ***Esc***.

Figure 037 – The space below the System Monitoring subtask, demonstrating what the window looks like when the timer and date are disabled.

Pressing the escape key a second time will reset the entire task and restore it to a "ready" state. At this point, one can simply restart the trial/practice, or load a new one. The "ready" state is indexed by the yellow color of timer on the task (if visible). PLEASE NOTE: All performance data from the previous run will be erased once the task has gone from a "suspended" to "ready" state. If you would like a copy of the data before it is erased, make sure to copy the data folder to another directory before bringing the task back to a "ready" state.

The figure above (Figure 038) shows an example of the previously illustrated practice going from a suspended state to a "ready" state.

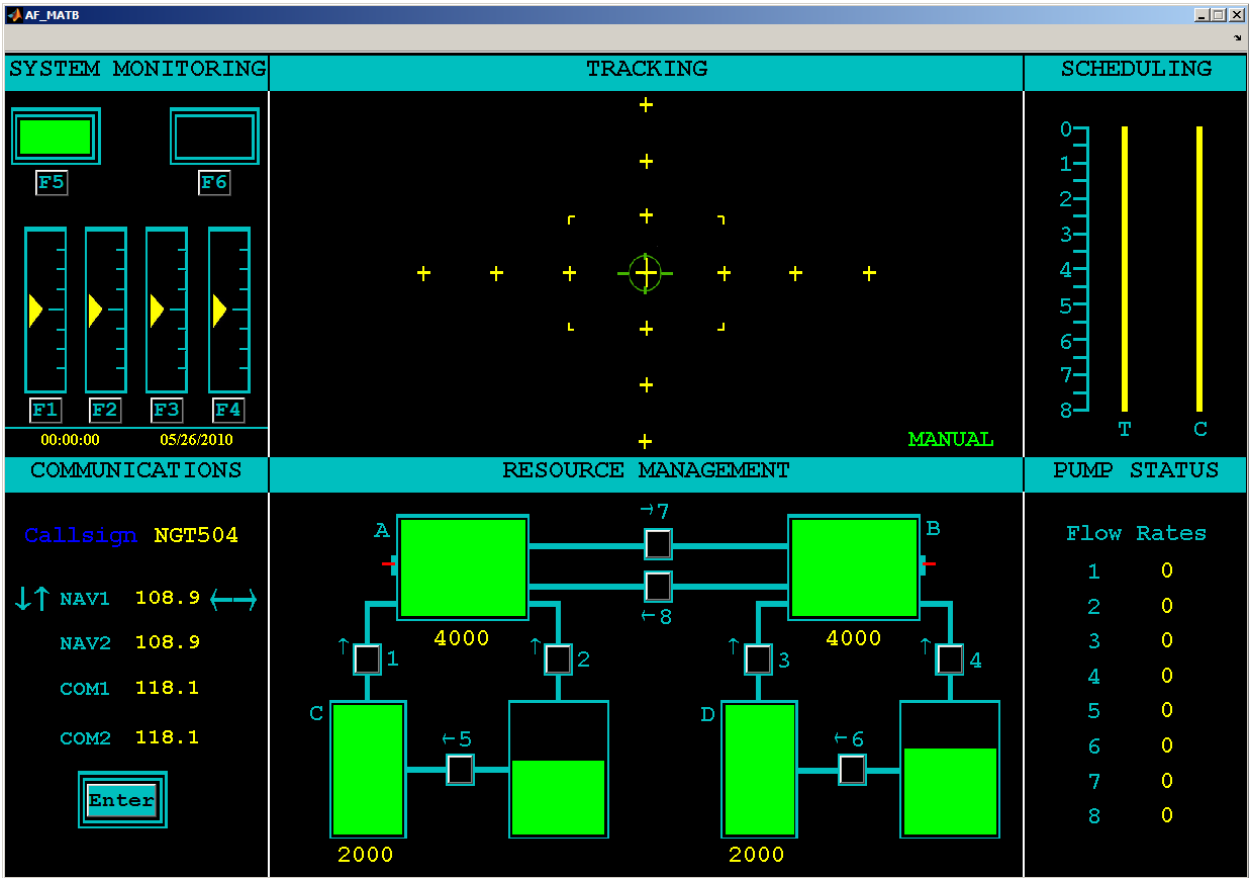


Figure 038 – The AF_MATB window after ***Esc*** was pressed again, placing the task in the "ready" state.

Pause | Break
If this option is enabled, it will allow for suspension of the task, practice, and all processes associated with the task for that particular run. Pressing the ***Pause*** key once will suspend the processes of the task. The "paused" state is indexed by a green color of the timer on the task. Pressing this key again will resume all processes from the exact point at which they were left and will also restore the timer to its normal, yellow color.

Please note that if the **Enable Pausing?** parameter is disabled, then pressing this button will do nothing. Also, please note that if the **Display Date and Time?** parameter is disabled, then the green "Paused" text in the task window will not appear, and the task will appear frozen, just as it does when the **Display Date and Time?** parameter is disabled and an individual presses ***Esc (Escape)*** (see Figure 039).

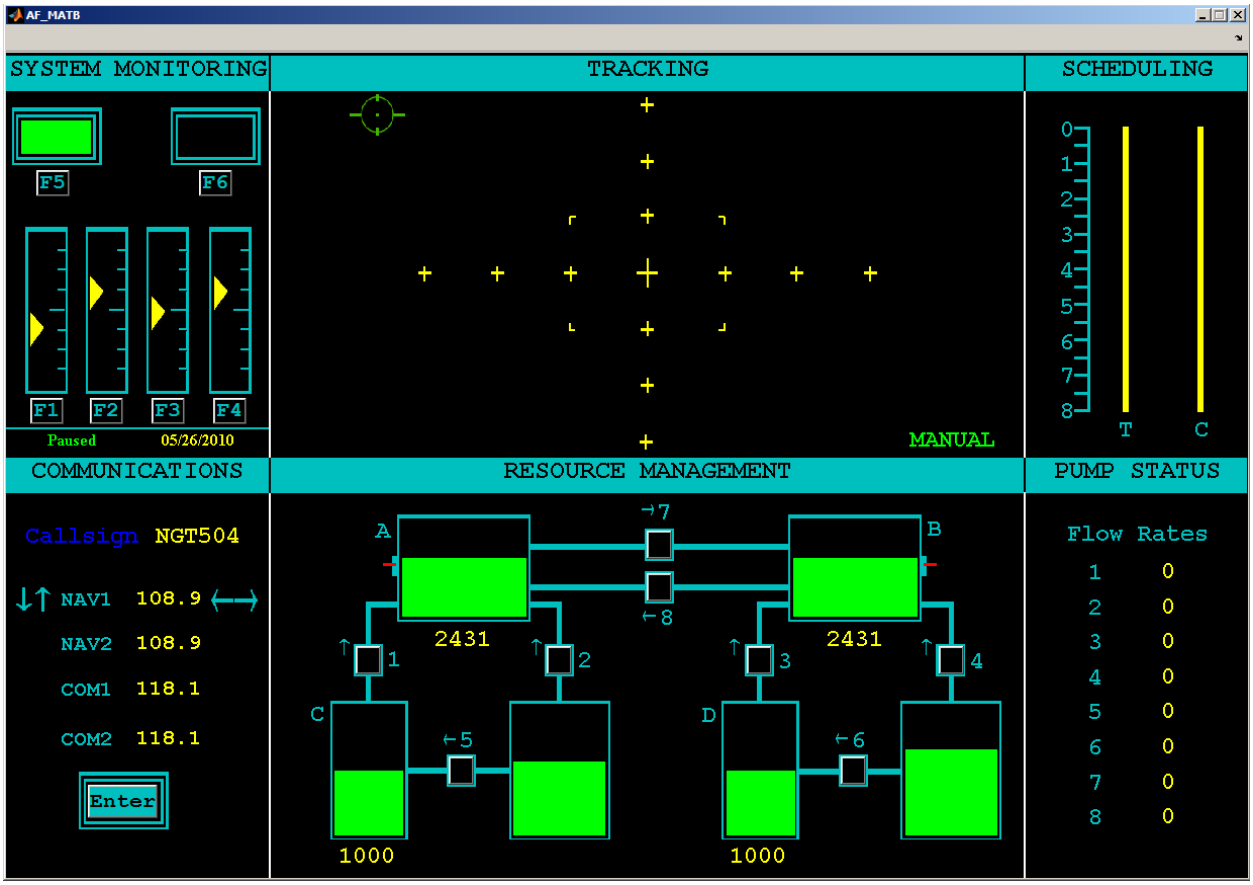


Figure 039 – What the task looks like when paused.

Home

The “New Load” button. During a run, a paused state, or a ready state, pressing this button will prompt a load screen and allow the user to select a new script (see Figure 040). Loading a new script will delete and replace all previous data, so this should not be used if your goal is to acquire performance data from multiple runs. However, if the goal is to expose the user to many different scenarios in short succession, such as for training purposes, this key will work well.

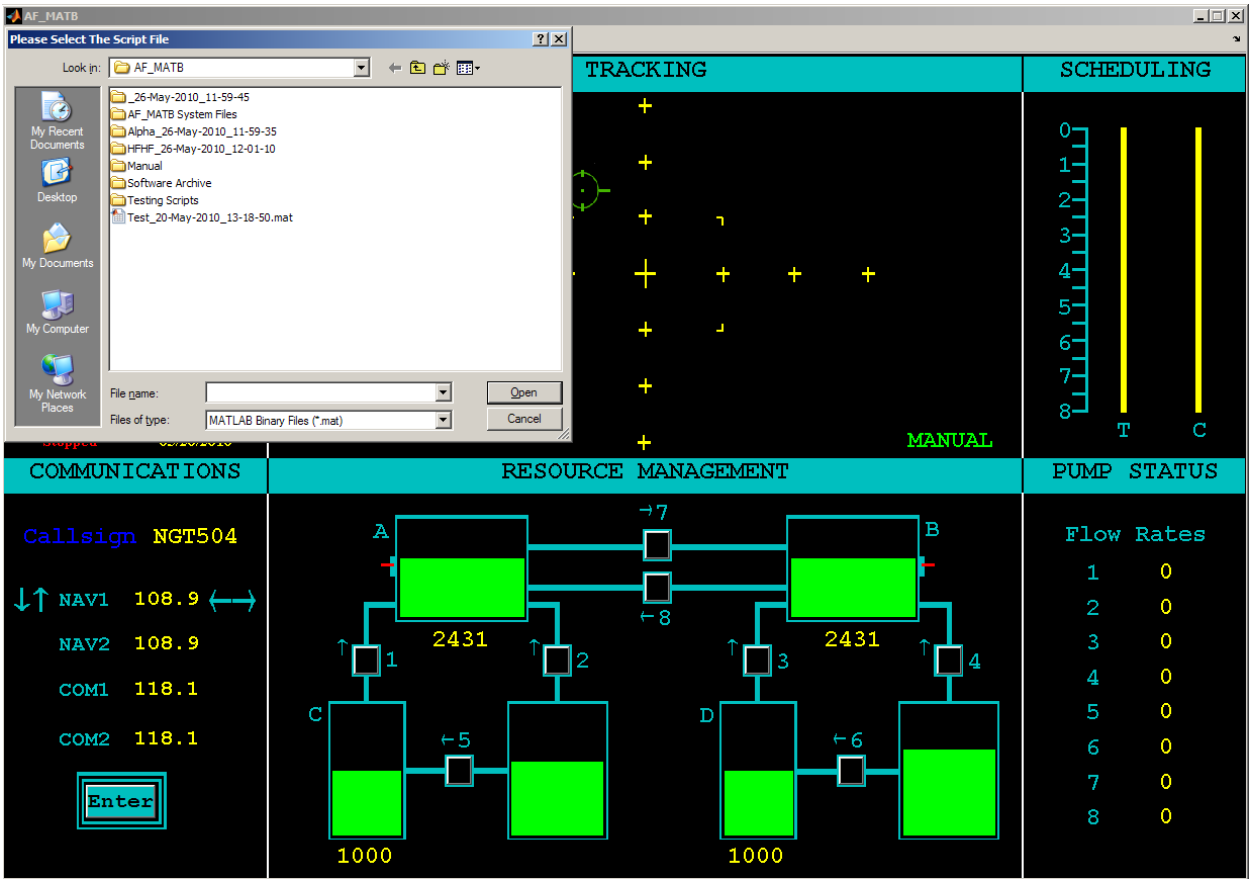


Figure 040 – The Task window with script loading window after pressing ***Home***.

If the loading process is interrupted, if the loading screen is closed, or if the file you were attempting to load is corrupted, then you will see the message showing in the following figure (Figure 041).

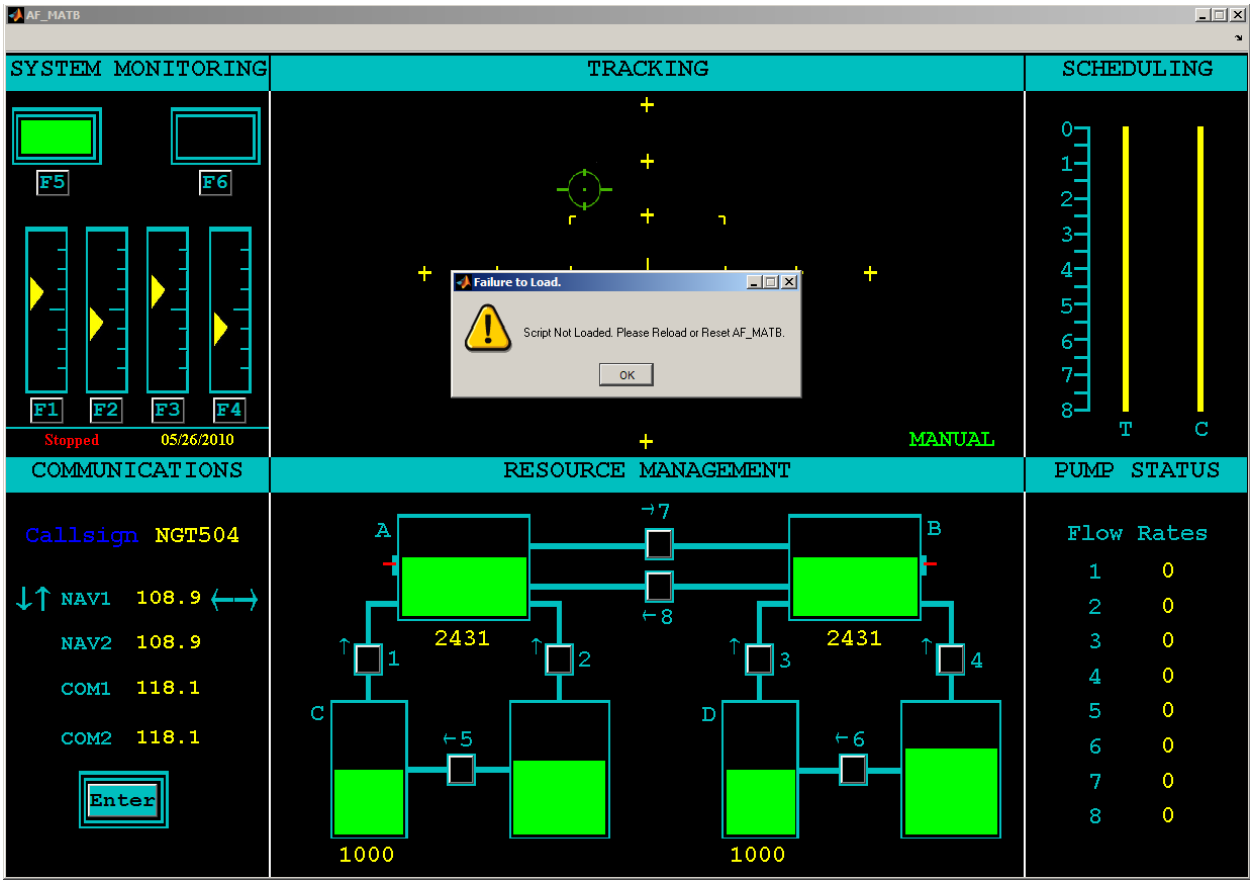


Figure 041 – Failure to load a script.

If loading is successful, then you will see the message indicated in the following figure (Figure 042). Please note that you can load custom parameters as well as full scripts. In the example below, a custom set of parameters was loaded to disable display of the date and time.

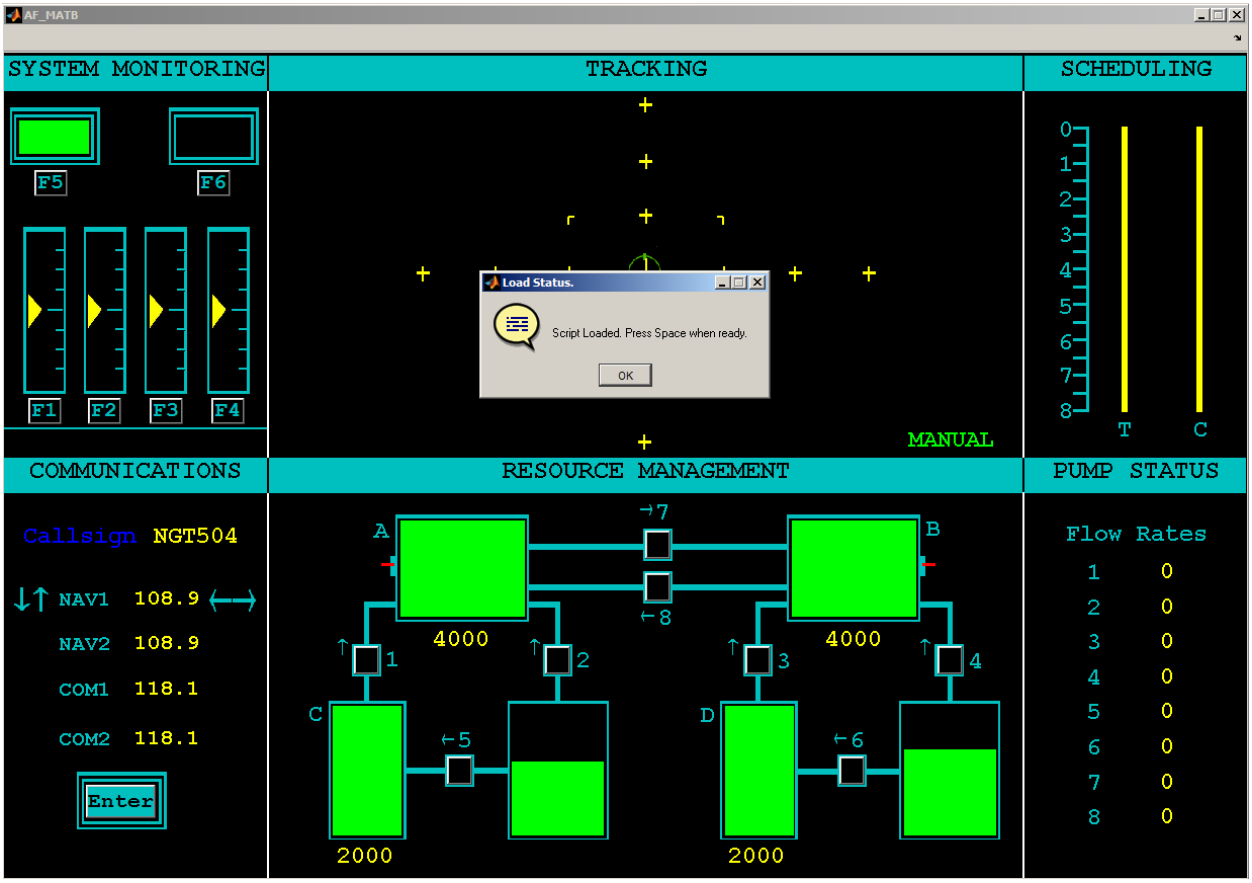


Figure 042 – Script successfully loaded.

4.4. Event-Related Commands

4.4.1. Transition Logging

Q

Allows the user to indicate precisely when they feel that the difficulty of the task has changed. Pressing the „Q" key on the keyboard records the timestamp, as well as the previous and current trials. This data is output into a file that also contains a list of all of the actual trial transitions in a given script, so that the comparison between the user"s perceptions and the actual, programmed task difficulty can be made quickly.

If this key is pressed and no actual script is present, then AF_MATB will log the action in the Master Event Log and nowhere else.

4.4.2. System Management

Shift + F1

Allows the user or experimenter to manually trigger a fault in Gauge 1. Gauge 1 will continue to malfunction until it is corrected or until the timeout signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + F2

Allows the user or experimenter to manually trigger a fault in Gauge2. Gauge 2 will continue to malfunction until it is corrected or until the timeout signal is been sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + F3

Allows the user or experimenter to manually trigger a fault in Gauge 3. Gauge 3 will continue to malfunction until it is corrected or until the timeout signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + F4

Allows the user or experimenter to manually trigger a fault in Gauge 4. Gauge 4 will continue to malfunction until it is corrected or until the timeout signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + F5

Allows the user or experimenter to manually trigger a fault in Light 1 (Green Light). Light 1 will continue to malfunction until it is corrected or until the timeout signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + F6

Allows the user or experimenter to manually trigger a fault in Light 2 (Red Light). Light 2 will continue to malfunction until it is corrected or until the timeout signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

F7
Simulates a timeout of a Gauge 1 error. After pressing this key, Gauge 1 will resume normal functioning. Please note, that this key will not produce the same response as a correct identification of a Gauge 1 error, nor is it coded as a correct identification in the Master Event Log.

F8
Simulates a timeout of a Gauge 2 error. After pressing this key, Gauge 2 will resume normal functioning. Please note, that this key will not produce the same response as a correct identification of a Gauge 2 error, nor is it coded as a correct identification in the Master Event Log.

F9
Simulates a timeout of a Gauge 3 error. After pressing this key, Gauge 3 will resume normal functioning. Please note, that this key will not produce the same response as a correct identification of a Gauge 3 error, nor is it coded as a correct identification in the Master Event Log.

F10
Simulates a timeout of a Gauge 4 error. After pressing this key, Gauge 4 will resume normal functioning. Please note, that this key will not produce the same response as a correct identification of a Gauge 4 error, nor is it coded as a correct identification in the Master Event Log.

IMPORTANT: Due to the nature of the Windows Operating System, pressing *F10* is similar to pressing *Alt*, and the focus on the window will move away from the actual AF_MATB GUI and be redirected towards the Menu Bar (which is disabled). At this point, the task may appear unresponsive to other keyboard commands. Simply press *F10*, *Alt*, or click on the task window again to move focus back to AF_MATB.

F11
Simulates a timeout of a Light 1 error. After pressing this key, Light 1 will resume normal functioning. Please note, that while this key will produce the same end result as a correctly identified error, this key is not coded as one in the Master Event Log.

F12
Simulates a timeout of a Light 2 error. After pressing this key, Light 2 will resume normal functioning. Please note, that while this key will produce the same end result as a correctly identified error, this key is not coded as one in the Master Event Log.

4.4.3. Resource Management

Shift + 1

Allows the user or experimenter to manually trigger a fault in Pump 1. Pump 1 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 2

Allows the user or experimenter to manually trigger a fault in Pump 2. Pump 2 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 3

Allows the user or experimenter to manually trigger a fault in Pump 3. Pump 3 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 4

Allows the user or experimenter to manually trigger a fault in Pump 4. Pump 4 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 5

Allows the user or experimenter to manually trigger a fault in Pump 5. Pump 5 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 6

Allows the user or experimenter to manually trigger a fault in Pump 6. Pump 6 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 7

Allows the user or experimenter to manually trigger a fault in Pump 7. Pump 7 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Shift + 8

Allows the user or experimenter to manually trigger a fault in Pump 8. Pump 8 will continue to malfunction until it is corrected or until the timeout/fix signal is sent. This event is coded as a manually triggered error and will be indicated as such in the Master Event Log.

Alt + 1
Allows the experimenter or user to manually repair a Pump 1 error. After pressing this key, Pump 1 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log as a manual pump correction.

Alt + 2
Allows the experimenter or user to manually repair a Pump 2 error. After pressing this key, Pump 2 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 3
Allows the experimenter or user to manually repair a Pump 3 error. After pressing this key, Pump 3 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 4
Allows the experimenter or user to manually repair a Pump 4 error. After pressing this key, Pump 4 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 5
Allows the experimenter or user to manually repair a Pump 5 error. After pressing this key, Pump 5 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 6
Allows the experimenter or user to manually repair a Pump 6 error. After pressing this key, Pump 6 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 7
Allows the experimenter or user to manually repair a Pump 7 error. After pressing this key, Pump 7 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Alt + 8
Allows the experimenter or user to manually repair a Pump 8 error. After pressing this key, Pump 8 will resume normal functioning. Please note that while this key will produce the same result as a script-triggered error timeout, it is not coded as such in the Master Event Log. It will be properly indicated, in the Master Event Log, as a manual pump correction.

Please note, there is no specified timeout for all manually triggered errors. The task will continue to operate with the error until it is fixed or until a timeout signal is sent. Also, please note that all commands in **Event Related Commands** section, with the exception of those in **Transition Logging**, can be disabled via the **Allow Manual Event Triggering?** parameter.

4.5. Task Options

4.5.1. Tracking

Shift + T
Allows the user or experimenter to manually trigger the Tracking Task‟s Autopilot during the task. The Autopilot for the Tracking Task will remain on until the user/experimenter or currently loaded script restores the task to Manual operation.

Alt + T
Allows the user or experimenter to manually restore the Tracking Task to Manual operation. The Tracking Task will continue to remain in Manual mode until the user/experimenter or currently loaded script activates the Tracking Task‟s Autopilot.

Shift + L
Allows the user or experimenter to manually change the level of difficulty of the Tracking Task. Please note that the task difficulty can only be changed if the Tracking Task is in Manual Mode. Pressing these keys will change the difficulty of the task to “Low.”

Shift + M
Allows the user or experimenter to manually change the level of difficulty of the Tracking Task. Please note that the task difficulty can only be changed if the Tracking Task is in Manual Mode. Pressing these keys will change the difficulty of the task to “Medium.”

Shift + H
Allows the user or experimenter to manually change the level of difficulty of the Tracking Task. Please note, the task difficulty can only be changed if the Tracking Task is in Manual Mode. Pressing these keys will change the difficulty of the task to “High.”

Shift + I
Allows the user or experimenter to manually change the inversion of the joystick used in the Tracking Task. If the user prefers to have an inverted joystick, pressing these keys will toggle the inversion. By default, pulling back on the joystick will move the Tracking Target up the Y-Axis, and pushing forward on the joystick will move the Tracking Target down the Y-Axis.

4.5.2. Resource Management

Shift + F

Allows the user or experimenter to manually trigger the Resource Management Task‟s Autopilot during the task. The Resource Management Task will remain in Autopilot until the user/experimenter or currently loaded script restores the task to Manual mode.

Alt + F

Allows the user or experimenter to manually restore the Resource Management Task to Manual mode. The Resource Management Task will remain in Manual mode until the user/experimenter or currently loaded script activates the Resource Management‟s Autopilot.

Shift + A

Allows the user or experimenter to toggle the Resource Management Autopilot‟s “Hide” function. While the Autopilot is enabled, by default, the user is able to see the Autopilot turn pumps on and off. This may become distracting, so the “Hide” function was added to eliminate visual distraction while the Resource Management Autopilot was engaged. When enabled, all pumps will turn blue, and the flow rates for all pumps will be listed as “Auto.” This function can only be toggled while the Resource Management Autopilot is engaged.

5.0 RUNNING AF_MATB

When AF_MATB is executed, the first thing the task will ask for is a Subject ID. This will be included in the name of the folder that contains all of the performance files, as well as all performance files, themselves. AF_MATB will automatically read filenames and check to make sure that no invalid characters are entered. Press OK or the ***Enter*** key to submit your Subject ID (see Figure 043).

Please note that the user may want to include run information here as well. For example, instead of using "S01" as the Subject ID, you may want to put use "S01_R01," indicating that this is the first run for S01. This will make identification of Performance files easier.

After entering a valid Subject ID, AF_MATB will then determine if the information loaded into the task was loaded by clicking any of the …**Continue to AF_MATB** buttons in AF_MATB_Parameters or AF_MATB_ScriptGenerator.

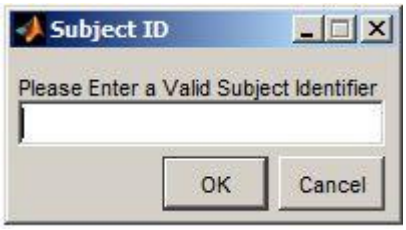


Figure 043 – The first dialog box you will see when executing AF_MATB, the Subject ID dialog.

In the below figure (Figure 044), information was loaded directly from AF_MATB_Parameters. As a result, the task will verify that the user still does not want to load a script file into the program. Please note that loading scripts that were not constructed using custom parameters may result in unstable task behavior and is not recommended. It is best to load a script designed using those custom parameters by using the AF_MATB_ScriptGenerator.

Figure 044 – If parameters are loaded directly from AF_MATB_Parameters, you will see this dialog box.

Figure 045 gives an example of the message that will be displayed when information is loaded directly from AF_MATB_ScriptGenerator. Once this message is acknowledged, you will be asked to load the AF_MATB System Files folder, which will be discussed later.

In the event that no information was directly loaded into the task, AF_MATB will recognize that no files have been loaded and still give the user the opportunity to do so. At this stage, AF_MATB can execute in one of two modes, either using a script or running freely. In order to train subjects on the task, it may be beneficial to first run the task in free-mode which will allow them to get a feel for the task, in general.

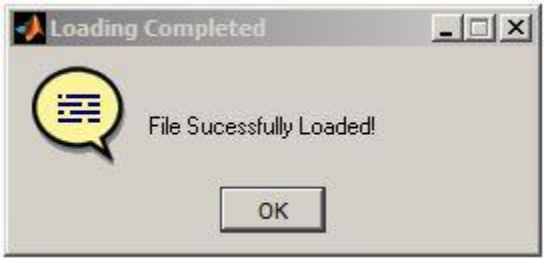


Figure 045 – When a file has been successfully loaded, this message will be displayed.

Figure 046 is an example of the message the user will receive if no script files have been loaded. If the user would like to operate in free-mode, instead of the scripted-mode, the user should simply choose "No" when this message appears. If the user answers "No," then they will be asked to identify where the AF_MATB System Files directory is located. This step will be covered in more detail in and is illustrated in Figure 050. If the user answers "Yes," they will be asked to select a script using the file selection utility.

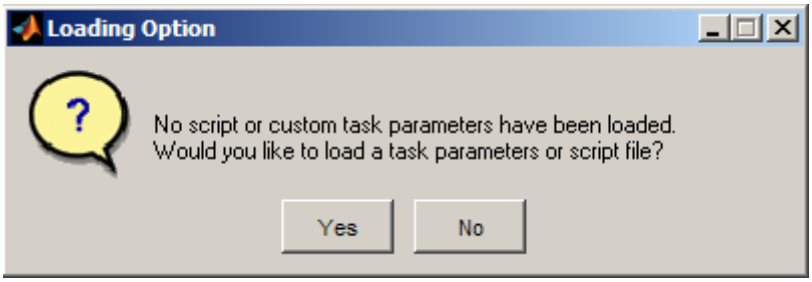


Figure 046 – Dialog box asking if you would like to load custom task parameters or a script.

If the experimenter indicates that they would like to load a task parameters or script file, the message above (Figure 047) will follow, which asks the user to choose what information they would like to load. Users have the option of loading a task parameters file, a script file, without any custom task parameters, or the entire script file, including event times and custom task parameters.

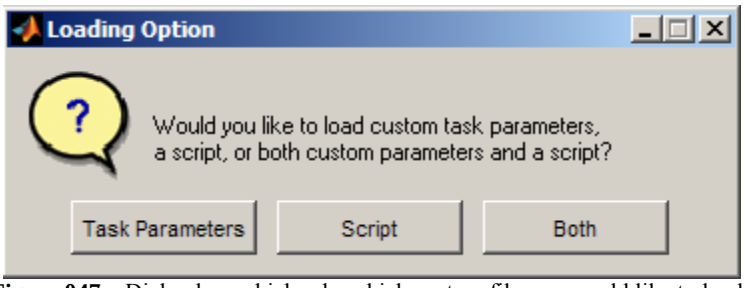


Figure 047 – Dialog box which asks which custom file you would like to load.

Regardless of which of the three loading options the user chooses, a file selection window will appear. The window, displayed above in Figure 048, can be used to navigate to the appropriate

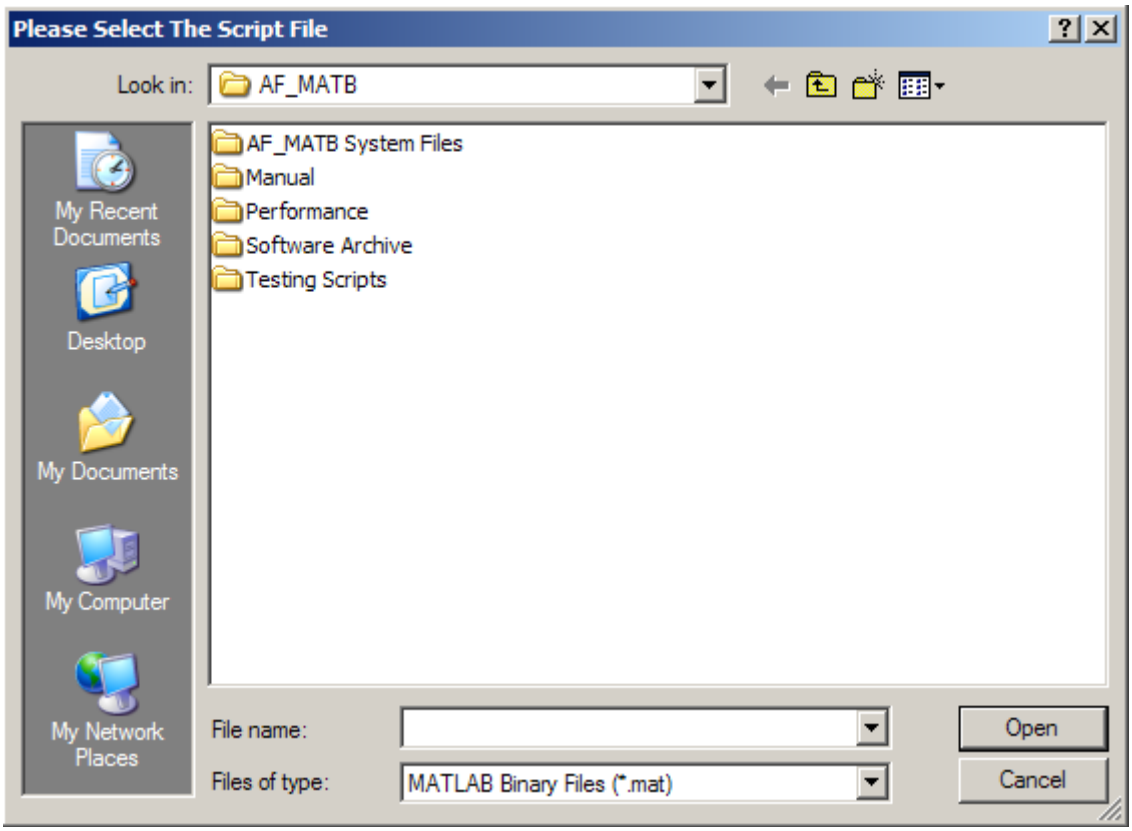


Figure 048 – Standard file selection utility in Windows which you can use to load custom parameters or scripts.

location where the script or task parameters file can be found. To open the file, the user should double-click on it or click on the file and then click "Open."

If loading is successful, you will see the message illustrated in Figure 045. If the user attempts to load a corrupt or invalid file, the message below (Figure 049) will appear.

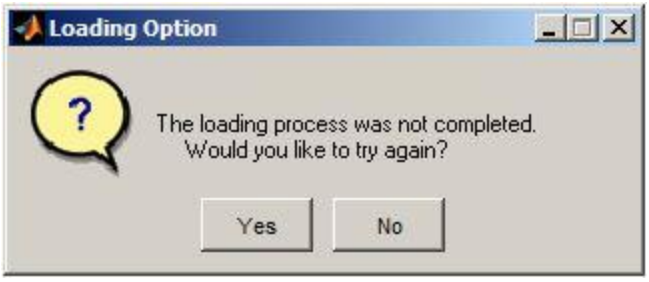


Figure 049 – Loading was not successful due to a corrupt script or parameters file.

Once all decisions have been made about the Subject ID and whether or not to load a script or custom task parameters, the user will be asked to identify the location of the AF_MATB System Files folder. As previously stated, this folder is critical to proper task operation and should not be modified in any way. Please refer to the figure below (Figure 050) for an example of how to use the directory selection tool. Simply click on the System Files folder and press "OK." The user should see "AF_MATB System Files" in the "Folder:" field in the window.

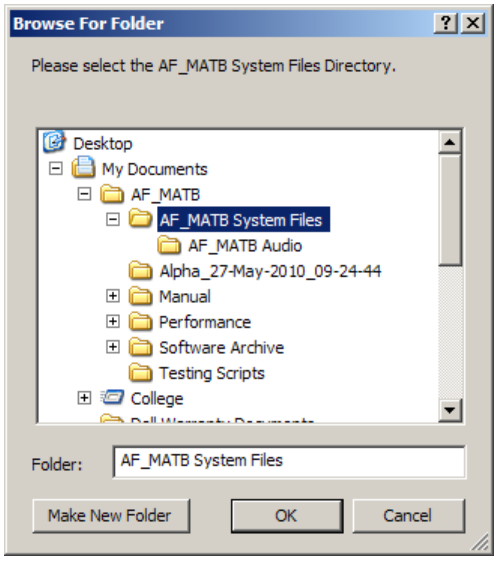


Figure 050 – Standard directory selection utility in Windows used to identify the location of the AF_MATB System Files folder.

Once the System Files folder has been successfully identified, the task window should appear with all parameters defined by any files loaded into it. (Obviously, some parameters that may have been set may not be visible, but parameters such as time/date display or callsign **Caller ID** should be correctly rendered.) **For the list of all files located in the AF_MATB System Files folder, please refer to Appendix A.**

At this point, the task is now ready to run. Press ***Space*** when you are ready to begin the run. Remember, there will be a five second counter that will appear before the task will begin normal operating behavior, illustrated in Figure 051.

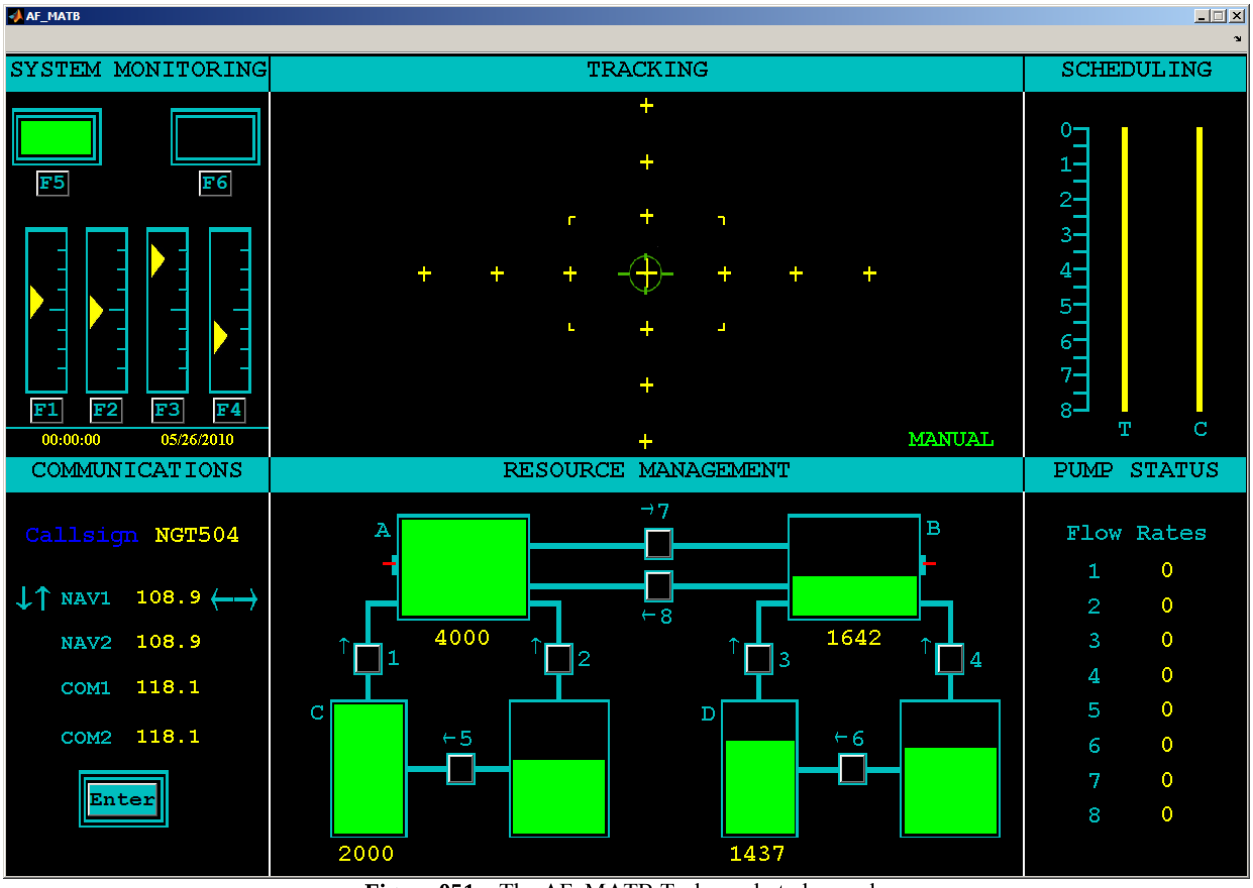


Figure 051 – The AF_MATB Task, ready to be used.

If a script has been loaded, then the task will run for as long as the script instructs. Once the script is completed, the task window will dispose and this message may appear (Figure 052), notifying the user that the program is in the process of saving information. The user should wait until this window disposes before doing anything else.

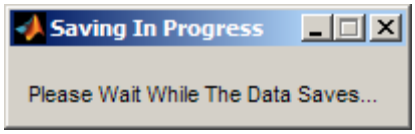


Figure 052 – Saving In Progress message, notifying the user to please wait until the data has finished saving.

5.1. Performance Folder Structure

Performance folders for an AF_MATB run are located in the directory from which the task was executed. In the folder, a run is broken down by trials. In AF_MATB, a run is defined as a session beginning when ***Space*** is pressed and ending when the "Saving In Progress" window appears and then disposes. Each run will generate a performance folder in the executing or current directory. However, in each performance folder, there may be multiple folders, as each run may be composed of multiple trials. A trial is defined as a **Low**, **Moderate**, **High Difficulty**, or **Transition Stage** in the Script Generator. Every line in the Condition List of the Script Generator signifies a trial, and as such, AF_MATB will parse performance and raw data by each trial. The task will also generate performance data across all trials and that data will be recorded in a Transition log and in a Master Event log. A digital copy of the raw performance will be generated as well, which can be used to go back and split performance into even smaller time series.

Figure 053 is an example of a performance folder. Please note that the performance folder was manually dragged into a folder called Performance; the performance folder was originally located in C:\Documents and Settings\millerwd\My Documents\AF_MATB.

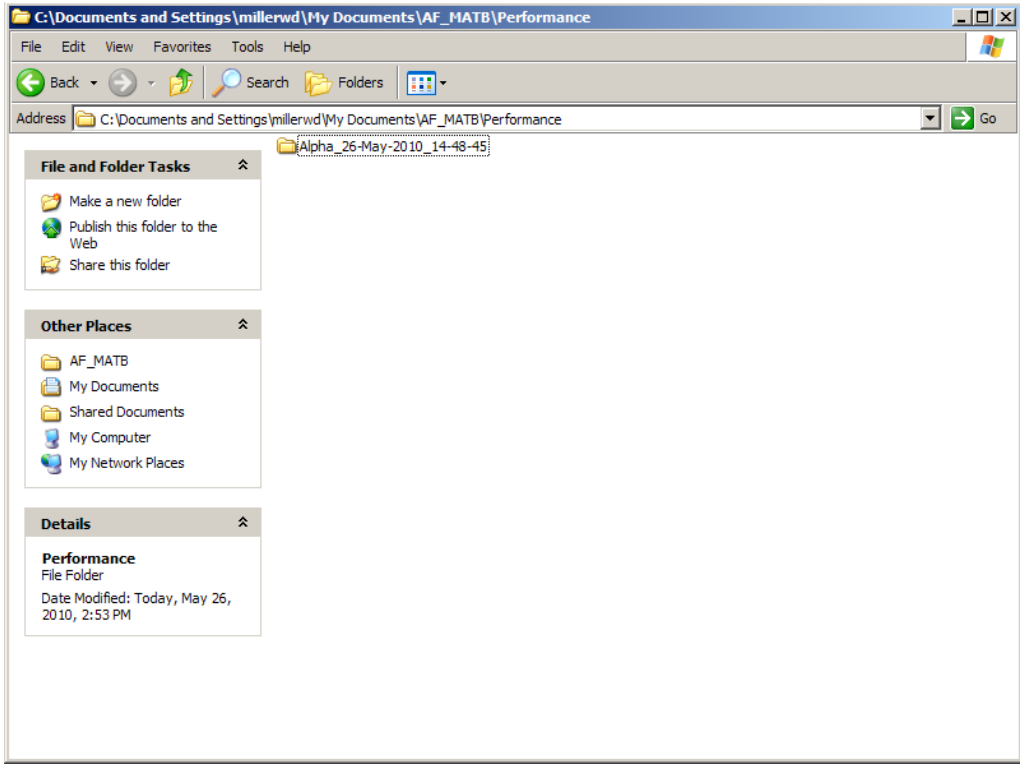


Figure 053 – An example performance folder generated after an experimental run.

Also note that in addition to the Subject ID, "Alpha," a date and time stamp was appended to the end of the Subject ID. This was a safety precaution implemented to prevent data loss or overwriting. In this case, no run information was added to the Subject ID, so you would assume that this performance file is the only performance file for Participant Alpha. If you were

conducting multiple runs on Alpha, you may want to ensure your Subject ID contains run information.

The figure below (Figure 054) displays the contents of the performance folder from the previous example. In this example, it is evident that there were two trials in this run. Note the four files not in folders. These files contain data for the entire run, and as such, were left in the root of the performance folder; they were not organized into one of the trial folders.

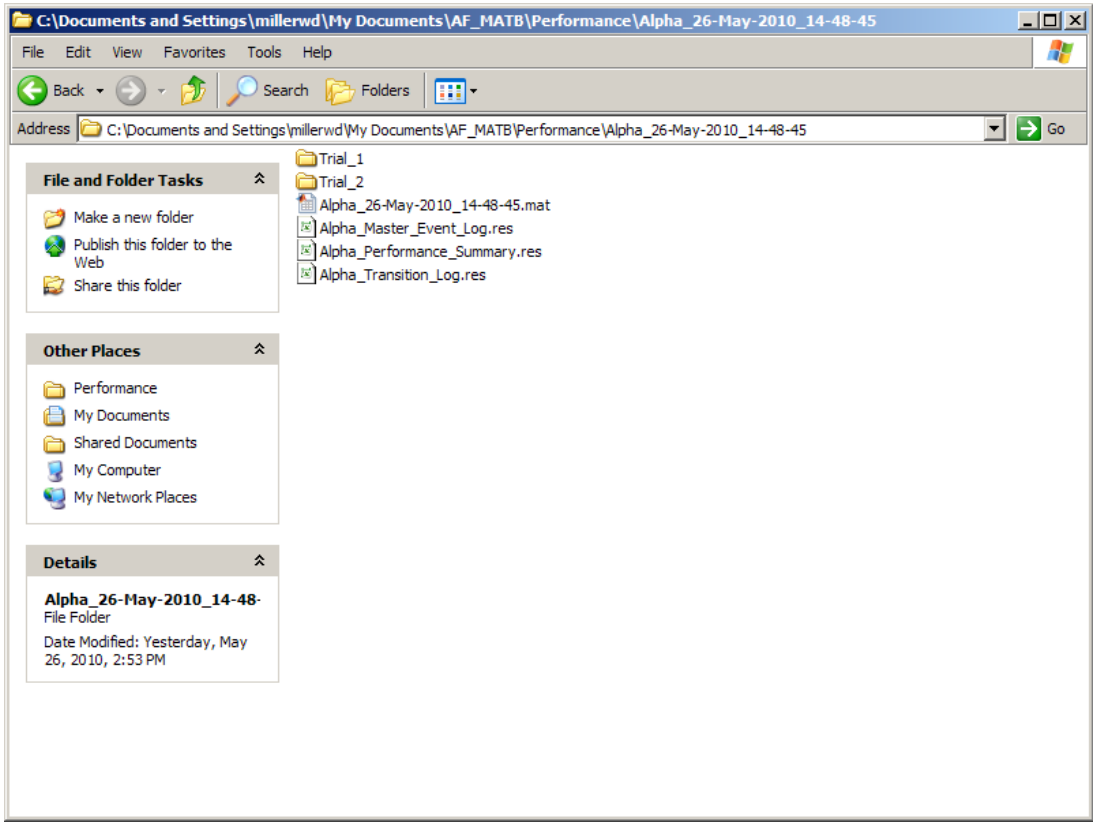


Figure 054 – The contents of the performance folder generated in **Figure 053**.

In addition to the four files in the root of the performance folder, there are a total of nine files in each trial folder. Each of these nine files contains different information about user performance on the task.

Below (Figure 055) is an example of all the files that are in a trial folder. Each file‟s name contains the Subject ID, followed by the file description, and finally, a number. The number is used to signify the trial that this file is for, in case multiple trials‟ performance files are mixed together. Thus, for a file from Trial_1, the end of the filename would be “_1.res,” while the same file from Trial_2 would be “_2.res.”

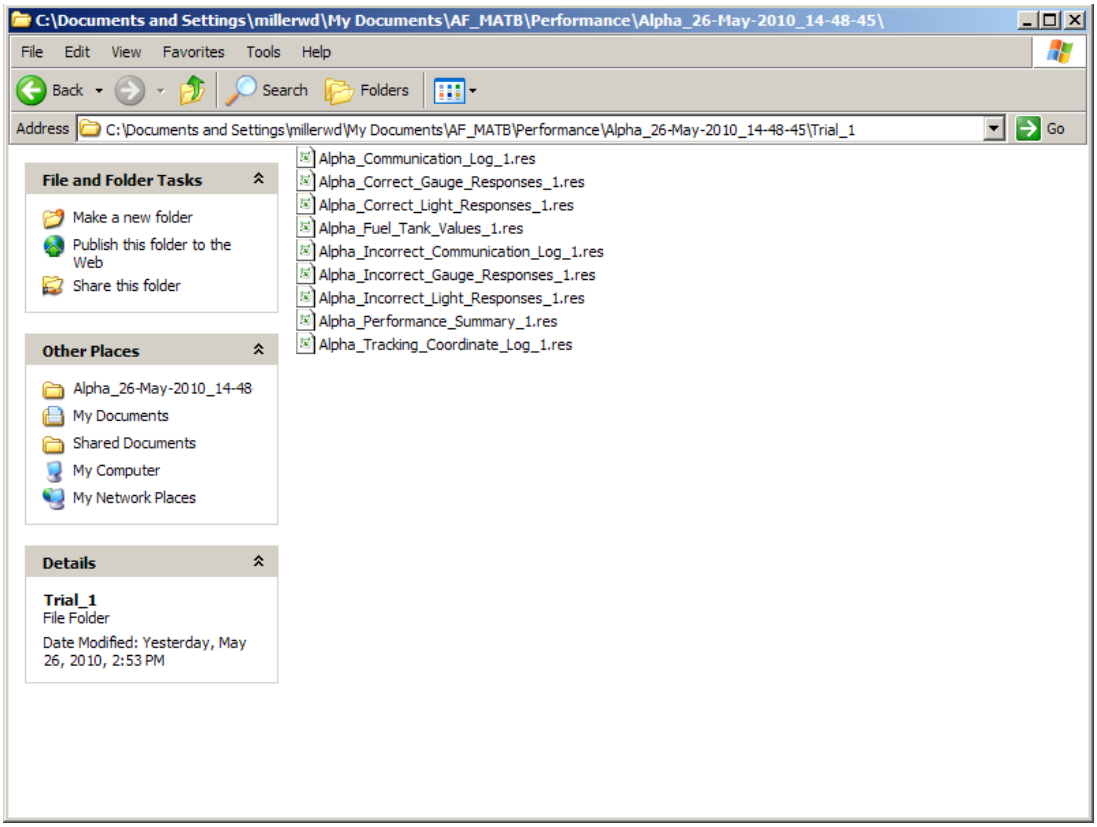


Figure 055 – The contents of one of the **Trial** folders illustrated in **Figure 054**.

In regards to file formats, all files in an experimental run are in one of two formats. For each experimental run, there will be one .mat file, which is the file that contains digital raw data for post processing, and a number of .res files. These .res files, the number of which will vary based on the number of trials in a run, can be opened using any text editor or using Excel. In order to determine exactly how many .res files a run will produce, multiply the number of trials by nine and then add three.

5.2. Performance File Descriptions

Figure 055 on the previous page, provided examples of the performance files one would find in a folder. The information in the following pages will name each file, identify its purpose and describe what data it contains.

5.2.1. Master Event Log

Contains all of the actions performed by the user as well as any scripted action the task carried out. This file is the only file that is generated regardless of whether a script file is loaded or not.

Specifically, the Master Event Log contains the Event Time, the Task or Item, and the Action that occurred. The Task or Item explains what subtask or component an action was applied to, and the Action describes exactly what behavior was performed.

5.2.2. Transition Log

Many of the runs in AF_MATB will be scripted to run several trials, each having a different level of difficulty. The purpose of the Transition Log is to record the types of trials contained in each run and to note the time that each trial begins and ends. The Transition Log is also set up to record the user‟s responses to perceived changes in the task‟s difficulty (which they designate by pressing the ***Q*** key). Thus, the Transition Log makes it possible for the researcher to compare the actual changes in task difficulty with the changes the user perceived.

The information in the log includes: each trial‟s level of difficulty, the time that each trial began and ended, and the difficulty of the trial that came before it. Additionally, there is a column called “Perceived or Actual Transitions.” If the information is labeled “Actual,” then it refers to a scripted transition, and if it is labeled “Perceived,” then it refers to an instance when the user perceived a change in task difficulty.

5.2.3. Communication Log

Logs all of the possibly correct actions the user makes in regards to the Communications Task.

Specifically, the Communication Log contains the Event Time that an auditory communication was played, its Correct Channel and Correct Frequency, and it‟s Target Signal, which describes whether the communication was a true communication (1) or a false communication (0). It also includes any Timeouts that occur, which are responses made outside of the timeframe designated for a given communication signal. . Finally, when the user responds to a communication, The Communication Log also records the Response Time, the Response Channel, and the Response Frequency.

5.2.4. Correct Gauge Responses

Logs all of the instances when the user correctly identifies a gauge fault in System Monitoring subtask.

Specifically, it contains the Event Time, the Response Time, the number of the gauge that the individual correctly identified as malfunctioning, and the Event Timeout, or time at which the gauge was automatically restored to normal functioning.

5.2.5. Correct Light Responses

Logs all of the correct light fault identifications for the Light component of the System Monitoring subtask.

Specifically, it contains the Event Time, the Response Time, the number of the light that the individual correctly identified as malfunctioning, and the Event Timeout, or time at which the light was automatically restored to normal functioning.

5.2.6. Fuel Tank Values

Logs all of the Fuel values for Tanks A and B for each trial. Values are logged at the rate which the task cycles in the computer, approximately every .1 seconds.

Specifically, it contains the Fuel Target, which is the target value for the tanks. This is provided because it can either be a default value or custom parameter and is required for manual calculation of the RMS. Furthermore, the log contains the Event Time, which is the time that information was logged, and the tank volumes, referred to as Tank A Value and Tank B Value. Finally, this file contains the calculations used to determine the Resource Management RMS, including the Tank A Difference and Tank B Difference, or the difference between those tanks and the Fuel Target, as well as the Tank A Deviation and Tank B Deviation, which are the squares of the Tank A and Tank B Differences, respectively.

5.2.7. Incorrect Communication Log

Logs all clearly incorrect actions the user makes in regards to the Communications Task for each trial.

Specifically, this log contains the Response Time when the incorrect response to a communication was made, as well as the Channel and Frequency used in the response.

5.2.8. Incorrect Gauge Responses

Logs all incorrect actions the user makes in regards to the Gauge component of the System Monitoring subtask for each trial.

Specifically, this log contains the Response Time and Gauge Number for each incorrect response.

5.2.9. Incorrect Light Responses

Logs all incorrect actions the user makes in regards to the Light component of the System Monitoring subtask for each trial.

Specifically, it contains the Response Time and Light Number for each incorrect response.

5.2.10. Tracking Coordinate Log

Logs all of the Tracking values for each trial. Values are logged at the rate which the task cycles, approximately every .1 seconds.

Specifically, this log contains the Center Coordinate, which is the X and Y pixel values for the center of the Tracking task. The Center Coordinate is used by AF_MATB to calculate the Tracking RMS and is provided for manual calculation if needed. Furthermore, it contains the Event Time, or logged time, that the X Coordinate and Y Coordinate pixel locations on the tracking crosshair were taken. Finally, it contains the calculations involved to determine the Tracking RMS, such as the X-Axis Difference and Y-Axis Difference from the X Coordinate and Y Coordinate of the tracking crosshair, respectively, as well as the X-Axis Deviation and Y-Axis Deviation, which are the squares of the X-Axis and Y-Axis Difference, respectively.

5.2.11. Performance Summary

Contains condensed performance numbers for all of the subtasks for MATB.

<u>System Monitoring</u>

Performance metrics for System Monitoring include event occurrences, correct responses, timed-out events, and incorrect responses (or false alarms). In addition, the mean and standard deviation of the reaction time of correct response are also provided. These metrics are broken down on several levels: the individual component level, such as the information for Gauge 1, the

total task level, such as the information for all Gauges combined, and the complete subtask, where all information from each gauge and light is combined to form the metrics for the entire System Management subtask.

Resource Management

Resource Management performance can be obtained by taking the RMS error from Tank A and summing that value with the RMS error of Tank B to get a total subtask metric. The RMS error involves two key parameters, the RMS Target Value, which specifies the target volume that the user should try to maintain in both Tanks A and B, and the RMS Interval, specifies the interval at which the tank volumes should be stored so that the task RMS Error can be calculated

Tracking

Tracking performance is also obtained by taking the RMS error from the center of the task window.

Communications

Communications performance is based simply on event occurrences.

The Total Communications metric looks at the total number of all communication signals played during a given trial, and is comprised of the sum of True Communications and False Communications.

The True Communications metric looks at the number of all communication signals addressed to the user‟s callsign during a given trial.

The False Communications metric looks at the number of all communication signals addressed to any callsign other than the callsign designated to the user.

The Correct Responses metric tells the experimenter the number of true communication signals that the user responded to correctly (with the correct channel and frequency).

The False Alarms metric tells the experimenter the number of false communication signals to which the user responded to with the correct channel and correct frequency.

Response Timeouts tell the experimenter the number of true communication signals that the user failed to respond to.

True Accuracy Errors look at the number of true communication signals that the user responded to with either the correct channel or correct frequency properly locked-in, but not both.

False Accuracy Errors look at the number of false communication signals that the user responded to with either the correct channel or correct frequency properly locked-in, but not both.

Unexplained Responses are described as user responses where both a frequency and channel were incorrectly locked-in, among other conditions in which the computer cannot account for a user‟s actions.

No Event Responses are when a frequency and channel were locked in when no communication signals were played during the task.

The Mean Correct Response Time is the average time it took for a user to correctly respond to the true communication signals.

The STD Correct Response Time is the standard deviation associated with the Mean Correct Response Time.

For examples of all 11 Performance Files, please refer to Appendix B.

6.0 AF_MATB PARAMETERS

One of the key points to AF_MATB is the fact that virtually every parameter used in the task can be manipulated to suit the experimenter. With this version, experimenters are afforded full control over virtually any and all options they may want to manipulate to suit their needs. In order to manipulate these parameters, a separate GUI was created, AF_MATB_Parameters, to gives the experimenter access to fifty task parameters in a user-friendly form to ensure ease of use, speed, and maximum functionality.

The parameters in AF_MATB can be divided into two important classes; script-critical parameters and operating-task parameters. Script-critical parameters are parameters that the Script Generator (known as AF_MATB_ScriptGenerator, discussed later) directly utilizes in order to properly create scripts with no logical errors. These parameters can include things like RMS Intervals or Task Timeouts and are directly tied to analyzing the performance of a user on the task. The second class, operating-task parameters, are critical to the proper function of the task, but do not play a role in the performance of the subject. Rather, these parameters are typically designed to manipulate the difficulty of the task on a fundamental level and afford the experimenter precise control over the task, such as how fast the gauges move, the starting value for each of the four tanks, and the user‟s callsign. From this point, each class can be further divided by the specific subtask that each parameter applies to (see Figure 056).

From this point on, we will discuss what each parameter does, acceptable values to enter, the default value of each parameter, and a picture of that parameter‟s field Some values for the parameters, while logically acceptable, may produce odd behavior in the task and should be avoided. The range/type of values that are acceptable for each parameter will be discussed later.

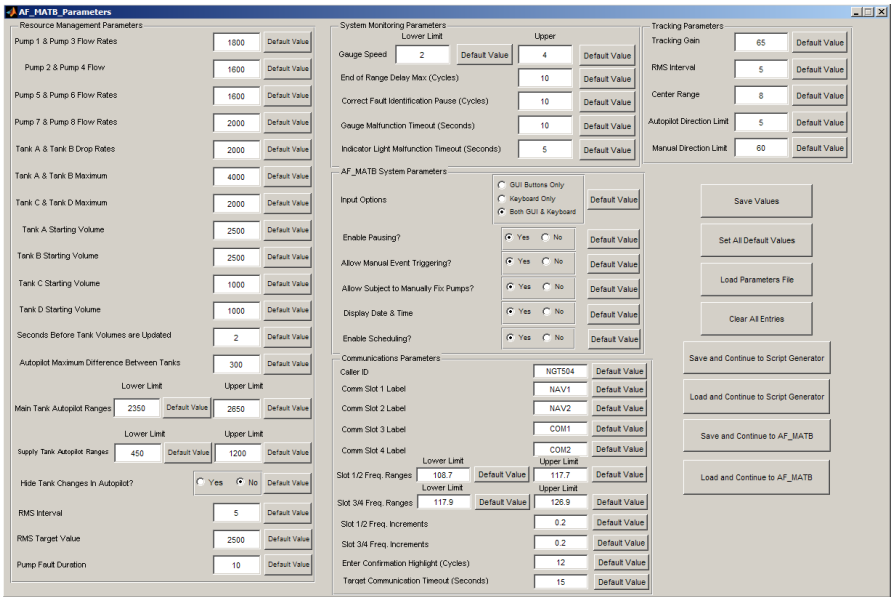


Figure 056 – AF_MATB_Parameters window when launched.

7.0 SYSTEM REQUIREMENTS

Hardware requirements for AF_MATB_Parameters include 1 GB of RAM, 2 GHz dual-core processor, a keyboard and mouse, and a 15-inch monitor (1280x1024 resolution), at minimum. It is important that these minimum requirements be met to ensure proper function of the utility.

AF_MATB_Parameters was designed for computers operating on Windows XP SP3 or higher. If the task is executed using a MATLAB script, then version 2007B or later should be used. Otherwise, it is recommended that MATLAB Compiler Runtime 7.8 or later be installed.

Please ensure that your DPI settings are set to default values. If your DPI is set to custom values, you may have window distortion.

8.0 PARAMETER CLASSES

8.1. Script-Critical Parameters

8.1.1. Resource Management Parameters

Pump Fault Duration

This parameter specifies the exact amount of time that a pump's failure (indicated by a red pump) will last. The pump will remain disabled for precisely the amount of time specified, unless the user manually corrects the fault. Once the amount of time specified for the failure has passed, the pump will return to the "off" position.

The default value for this parameter is 10 (seconds) (see Figure 057).

Appropriate values for this parameter are real numbers greater than 0.

Figure 057 – Pump Fault Duration.

RMS Interval

This parameter specifies the interval at which the tank volumes should be stored so that the task RMS Error can be calculated. The value "5" means "every 5 seconds, log the values of the tanks."

The default value for this parameter is 5 (seconds) (see Figure 058).

Appropriate values for this parameter are integers greater than 0.

Figure 058 – RMS Interval.

RMS Target Value

This parameter specifies the target volume that the user should try to maintain in both Tank A and Tank B. This value is critical to the proper calculation of RMS Error for the tanks.

The default value for this parameter is 2500 (units) (see Figure 059).

Appropriate values for this parameter are real numbers greater than the minimum operating value of the autopilot and less than the maximum operating value of the autopilot. These values can be found in the Operating-Task Parameters' Resource Management section, discussed later.

Figure 059 – RMS Target Value.

8.1.2. System Monitoring Parameters

Gauge Malfunction Timeout

This parameter specifies the amount of time that a gauge will operate in a malfunctioning range. After this amount of time, if the pump is not corrected, then it will automatically be restored to normal operating function.

The default value for this parameter is 10 (seconds) (see Figure 060).

Appropriate values for this parameter are real numbers greater than 0.

Figure 060 – Gauge Malfunction Timeout.

Indicator Light Timeout

This parameter specifies the amount of time that a light will indicate a malfunction. After this amount of time, if the light is not corrected, then it will automatically be restored to its normal indication.

The default value for this parameter is 5 (seconds) (see Figure 061).

Appropriate values for this parameter are real numbers greater than 0.

Figure 061 – Indicator Light Timeout.

8.1.3. Communications Parameters

Target Communication Timeout

This parameter specifies the amount of time that the user has to respond to a callsign addressed to him. After this amount of time, if no response is recorded, then it will scored as a Communications Response Timeout.

The default value for this parameter is 15 (seconds) (see Figure 062).

Appropriate values for this parameter are real numbers greater than 0.

Figure 062 – Target Communication Timeout.

8.1.4. Tracking Parameters

RMS Interval

This parameter specifies the interval at which the position of the crosshair should be stored so that the task RMS Error can be calculated. The value "5" means "every 5 seconds, log the position of the crosshair."

The default value for this parameter is 5 (seconds) (see Figure 063).

Appropriate values for this parameter are integers greater than 0.

Figure 063 – RMS Interval.

8.2. Task-Operating Parameters

8.2.1. Resource Management Parameters

Pumps 1 & 3 Flow Rates

This parameter specifies the amount of fuel that pumps 1 & 3 will pump into Tanks A and B and out of Tanks C and D per minute.

The default value for this parameter is 1800 (units/minute) (see Figure 064).

Appropriate values for this parameter are real numbers greater than 0.

Figure 064 – Pumps 1 & 3 Flow Rates.

Pumps 2 & 4 Flow Rates

This parameter specifies the amount of fuel that pumps 2 & 4 will pump into Tanks A and B per minute.

The default value for this parameter is (1600 units/minute) (see Figure 065).

Appropriate values for this parameter are real numbers greater than 0.

Figure 065 – Pumps 2 & 4 Flow Rates.

Pumps 5 & 6 Flow Rates

This parameter specifies the amount of fuel that pumps 5 & 6 will pump into Tanks C and D per minute.

The default value for this parameter is (1600 units/minute) (see Figure 066).

Appropriate values for this parameter are real numbers greater than 0.

Figure 066 – Pumps 5 & 6 Flow Rates.

Pumps 7 & 8 Flow Rates

This parameter specifies the amount of fuel that Pump 7 will pump from Tank A to Tank B, or that amount of fuel that Pump 8 will pump from Tank B to Tank A, per minute.

The default value for this parameter is 2000 (units/minute) (see Figure 067).

Appropriate values for this parameter are real numbers greater than 0.

Figure 067 – Pumps 7 & 8 Flow Rates.

Tanks A & B Drop Rate

This parameter specifies the amount of fuel Tanks A and B will lose per minute.

The default value for this parameter is 2000 (units/minute) (see Figure 068).

Appropriate values for this parameter are real numbers greater than 0.

Figure 068 – Tank A & B Drop Rates.

Tanks A & B Maximum

This parameter specifies the maximum volume of Tanks A and B.

The default value for this parameter is 4000 (units) (see Figure 069).

Appropriate values for this parameter are real numbers greater than 0.

Figure 069 – Tank A & B Maximum.

Tanks C & D Maximum
This parameter specifies the maximum volume of Tanks C and D.

The default value for this parameter is 2000 (units) (see Figure 070).

Appropriate values for this parameter are real numbers greater than 0.

Figure 070 – Tank C & D Maximum.

Tank A Starting Volume
This parameter specifies the volume that Tank A will start with when the task begins.

The default value for this parameter is 2500 (units) (see Figure 071).

Appropriate values for this parameter are integers greater than 0.

Figure 071 – Tank A Starting Volume.

Tank B Starting Volume
This parameter specifies the volume that Tank B will start with when the task begins.

The default value for this parameter is 2500 (units) (see Figure 072).

Appropriate values for this parameter are integers greater than 0.

Figure 072 – Tank B Starting Volume.

Tank C Starting Volume
This parameter specifies the volume that Tank C will start with when the task begins.

The default value for this parameter is 1000 (units) (see Figure 073).

Appropriate values for this parameter are integers greater than 0.

Figure 073 – Tank C Starting Volume.

Tank D Starting Volume
This parameter specifies the volume that Tank D will start with when the task begins.

The default value for this parameter is 1000 (units) (see Figure 074).

Appropriate values for this parameter are integers greater than 0.

Figure 074 – Tank D Starting Volume.

Seconds Before Tank Values are Updated
This parameter specifies the amount of time elapsed between visual updates of the tank volumes. This applies to both the pictorial representation of the tanks as well as the volume labels.

The default value for this parameter is 2 (seconds) (see Figure 075).

Appropriate values for this parameter are real numbers greater than 0.

Figure 075 – Seconds Before Tank Values are Updated.

Autopilot Maximum Difference Between Tanks

This parameter is used exclusively for the Fuel Autopilot mode. When the Fuel Autopilot is enabled, this parameter sets the threshold for when exactly the Autopilot should use Pumps 7 & 8. If the difference between Tank A and Tank B is greater than 300 units, the appropriate transfer pump will be used to shift fuel from Tank A to Tank B or vice versa.

The default value for this parameter is 300 (units) (see Figure 076).

Appropriate values for this parameter are real numbers greater than 0.

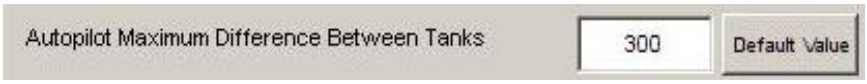


Figure 076 – Autopilot Maximum Difference Between Tanks.

Main Tank Autopilot Ranges

Lower Limit

This parameter is a setting used exclusively for the Fuel Autopilot mode. If Fuel Autopilot is enabled, the autopilot needs a threshold set for when exactly to use Pumps 1-4. This parameter states that if the volume of Tank A falls below this value, then Pumps 1 & 2 will turn on. If the volume of Tank B falls below this value, then Pumps 3 & 4 will turn on. This parameter is designed to help decrease some of the pump activity, as pumps constantly turning on and off may be distracting.

The default value for this parameter is 2350 (units) (see Figure 077).

Appropriate values for this parameter are real numbers greater than 0 but less than the previously defined **Tank A & B Maximum**.

Upper Limit

This parameter is a setting used exclusively for the Fuel Autopilot mode. If Fuel Autopilot is enabled, the autopilot needs a threshold set for when exactly to use Pumps 1-4. This parameter states that if the volume of Tank A is above the set threshold, then Pumps 1 & 2 will turn off. If the volume of Tank B is above the set threshold, then Pumps 3 & 4 will turn off. This parameter is designed to help decrease some of the pump activity, as pumps constantly turning on and off may be distracting.

The default value for this parameter is 2650 (units) (see Figure 077).

Appropriate values for this parameter are real numbers greater than 0 but less than the previously defined *Lower Limit*.

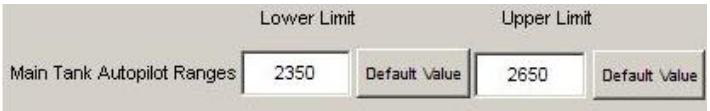


Figure 077 – Main Tank Autopilot Ranges.

Supply Tank Autopilot Ranges

Lower Limit

This parameter is a setting used exclusively for the Fuel Autopilot mode. If Fuel Autopilot is enabled, the autopilot needs a threshold set for when exactly to use Pumps 5 & 6. This parameter states that if the volume of Tank C falls below this value, then Pump 5 will turn on. If the volume of Tank B falls below this value, then Pump 6 will turn on. This parameter is designed to help decrease some of the pump activity, as pumps constantly turning on and off may be distracting.

The default value for this parameter is 450 (units) (see Figure 078).

Appropriate values for this parameter are real numbers greater than 0 but less than the previously defined **Tank C & D Maximum**.

Upper Limit

This parameter is a setting used exclusively for the Fuel Autopilot mode. If Fuel Autopilot is enabled, the autopilot needs a threshold set for when exactly to use Pumps 5 & 6. This parameter states that if the volume of Tank C is above this value, then Pump 5 will turn off. If the volume of Tank D is above this value, then Pump 6 will turn off. This parameter is designed to help decrease some of the pump activity, as pumps constantly turning on and off may be distracting.

The default value for this parameter is 1200 (units) (see Figure 078).

Appropriate values for this parameter are real numbers greater than 0 but less than the previously defined *Lower Limit*.

Figure 078 – Supply Tank Autopilot Ranges.

Hide Tank Changes in Autopilot

This parameter is a setting used exclusively for the Fuel Autopilot mode. The „Hide Tank Changes in Autopilot" option can be used to completely mask pump activities (which may be perceived as distracting). It turns all pumps the same color as the blue "piping" that connects them to the tanks and changes all Pump Status "Flow Rates" to "Auto." This setting achieves the same effect as the ***Shift + A*** command discussed in the *Keyboard Commands' Task Options* section and can easily be toggled depending on the participant/experimenter's preference.

The default value for this parameter is *"No"* (see Figure 079).

Appropriate values for this parameter are *"Yes" & "No."*

Figure 079 – Hide Tank Changes in Autopilot.

8.2.2. System Monitoring Parameters

Gauge Speed

Lower Limit

The speeds of gauges are somewhat dependent on the speed of the computer on which the experimenter is running the task. As a result, the experimenter has access to the speed of the gauges in the event that they are moving too fast or too slow. Overall, bringing this number closer to zero will slow the speed of the pointer, while bringing it closer to the *Upper Limit* will increase the speed of the pointer.

The default value for this parameter is 2 (pixels/second) (see Figure 080).

Appropriate values for this parameter are real numbers greater than 0.

Higher Limit

The speeds of gauges are somewhat dependent on the speed of the computer on which the experimenter is running the task. As a result, the experimenter has access to the speed of the gauges in the event that they are moving too fast or too slow. This value determines the maximum speed at which the pointers will move. Increasing this value will increase the maximum speed the pointer can reach.

The default value for this parameter is 4 (pixels/second) (see Figure 080).

Appropriate values for this parameter are real numbers greater than the previously mentioned *Lower Limit*.

PLEASE NOTE: The movement of the pointers is based on a random speed within the defined range. By increasing the difference between the Upper and Lower Limits, the pointer will travel at a wider range of random speeds, whereas a smaller difference will result in a smaller range of speeds.

Figure 080 – Gauge Speed.

End of Range Delay Max (Cycles)

When the pointer on a gauge reaches the top and bottom of its moving range, including both the operating and malfunctioning ranges, the pointer will pause for a set number of cycles. This delay is influenced by the speed of the computer that AF_MATB is running on, so if the delay is not long enough, increase the limit accordingly.

The default value for this parameter is 10 (cycles) (see Figure 081).

Appropriate values for this parameter are integers greater than 0.

Figure 081 – End of Range Delay Max (Cycles).

Correct Fault Identification

When a gauge is malfunctioning and a user correctly identifies that malfunction, the pointer on the gauge is returned to the center, where it, as well as a small yellow rectangle, is frozen for a set amount of time. This parameter defines how long the pointer will be frozen in the middle and how long the yellow "correct-identification" indicator will be present before disappearing and allowing the gauge to resume normal function. This delay is influenced by the speed of the computer that AF_MATB is running on, so if the delay is not long enough, increase the limit accordingly.

The default value for this parameter is 10 (cycles) (see Figure 082).

Appropriate values for this parameter are integers greater than 0.

Figure 082 – Correct Fault Identification.

8.2.3. AF_MATB System Parameters

Input Options

This parameter controls what type of input the subject may utilize. All user inputs in AF_MATB are encoded for the keyboard, as well as the clickable icons (or buttons) in the GUI. The only task in AF_MATB that does not support multiple inputs is the Tracking task, which requires a joystick.

The default value for this parameter is *"Both GUI and Keyboard"* (see Figure 083).

Appropriate values for this parameter are *"Buttons Only," "Keyboard Only," "Both GUI and Keyboard."*

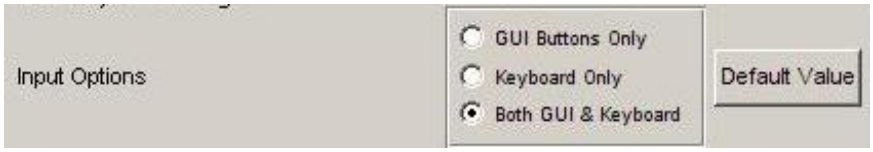


Figure 083 – Input Options.

Enable Pausing?

This parameter controls the ability to pause in the middle of a run. During training runs, pausing may be useful for teaching or emphasizing key points.

The default value for this parameter is *"Yes"* (see Figure 084).

Appropriate values for this parameter are *"Yes," "No."*

Figure 084 – Enable Pausing?.

Allow Manual Event Triggering?

This parameter controls the participant"s ability to manually trigger System Monitoring or Resource Management faults, as well as other keyboard code options.

The default value for this parameter is *"Yes"* (see Figure 085).

Appropriate values for this parameter are *"Yes," "No."*

Figure 085 – Allow Manual Event Triggering?.

Allow Subject to Manually Fix Pumps?
This parameter controls the participant"s ability to manually fix (remove) pump faults via the keyboard command.

The default value for this parameter is *"Yes"* (see Figure 086).

Appropriate values for this parameter are *"Yes," "No."*

Figure 086 – Allow Subject to Manually Fix Pumps?.

PLEASE NOTE: The **Input Options**, **Enable Pausing?**, **Allow Manual Event Triggering?** and **Allow Subject To Manually Fix Pumps?** parameters all affect various task response abilities. As a result, the following matrices will help clarify exactly which functions are or are not enabled based on the values of certain parameters. For details on keyboard commands, please refer to the task manual.

Table 1 details response abilities when the "GUI Buttons Only" parameter of **Input Options** is selected.

Table 1 – Breakdown of what keyboard commands are functional given the configuration of specific System Options for the input mode "*GUI Buttons Only."*

System Options	Event Triggering?	Yes	No	Yes	Yes	No	Yes	No	No
	Pump Fixing?	Yes	Yes	No	Yes	No	No	Yes	No
	Pausing?	Yes	Yes	Yes	No	Yes	No	No	No
Keyboard Commands	Shift + F1 – Shift F6	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	F7 - F12	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	Shift + 1 – Shift + 8	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	Alt + 1 – Alt + 8	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled	Enabled	Disabled
	Pause	Enabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled	Disabled
	F1 - F6	Disabled							
	1 - 8	Disabled							
	Escape	Enabled							
	Home	Enabled							
	Space	Enabled							

Table 2 details the response abilities when either of the other two parameters of **Input Options** is selected.

Table 2 – Breakdown of what keyboard commands are functional given the configuration of specific System Options for the input mode "*Keyboard Only"* or *"Both GUI and Keyboard."*

System Options	Event Triggering?	Yes	No	Yes	Yes	No	Yes	No	No
	Pump Fixing?	Yes	Yes	No	Yes	No	No	Yes	No
	Pausing?	Yes	Yes	Yes	No	Yes	No	No	No
Keyboard Commands	Shift + F1 – Shift F6	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	F7 - F12	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	Shift + 1 – Shift + 8	Enabled	Disabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled
	Alt + 1 – Alt + 8	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled	Enabled	Disabled
	Pause	Enabled	Enabled	Enabled	Disabled	Enabled	Disabled	Disabled	Disabled
	F1 - F6	Enabled							
	1 - 8	Enabled							
	Escape	Enabled							
	Home	Enabled							
	Space	Enabled							

Display Date and Time?
This parameter controls the display for the date, as well as the running time of AF_MATB in the task window. This was an option that was included in the original software design.

The default value for this parameter is *"Yes"* (see Figure 087).

Appropriate values for this parameter are *"Yes," "No."*

Figure 087 – Display Date and Time?.

Enable Scheduling?
This parameter activates the scheduling window in AF_MATB. By reading script-loaded parameters, when this option is enabled to *"Yes,"* red bars will appear on the Communication Timeline, each bar indicative of a communication event. Larger rectangles will appear in the Tracking Timeline, indicative of the length of the specific tracking difficulty. These bars are color-coded based on the tracking difficulty.

The default value for this parameter is *"Yes"* (see Figure 088).

Appropriate values for this parameter are *"Yes," "No."*

Figure 088 – Enable Scheduling?.

8.2.4. Communications Parameters

Caller ID

This parameter allows the experimenter to change the label of the target callsign in the communication task.

The default value for this parameter is "NGT504" (see Figure 089).

Appropriate values for this parameter are at least one character with a maximum of seven or eight characters.

PLEASE NOTE: This variable is purely cosmetic. It has no bearing on internal scoring procedures used to determine participant responses to communication signals. Manipulation of the Communication Task, such as the implementation of new Communication audio files, will most likely involve some manual recoding of AF_MATB.

Figure 089 – Caller ID.

Comm Slot 1 Label

This parameter allows the experimenter to change the label of the first communication slot in the communication task.

The default value for this parameter is "NAV1" (see Figure 090).

Appropriate values for this parameter are at least one character with a maximum of four or five characters.

PLEASE NOTE: This variable is purely cosmetic. It has no bearing on internal scoring procedures used to determine participant responses to communication signals. Manipulation of the Communication Task, such as the implementation of new Communication audio files, will most likely involve some manual recoding of AF_MATB.

Figure 090 – Comm Slot 1 Label.

Comm Slot 2 Label
This parameter allows the experimenter to change the label of the second communication slot in the communication task.

The default value for this parameter is "NAV2" (see Figure 091).

Appropriate values for this parameter are at least one character with a maximum of four or five characters.

PLEASE NOTE: This variable is purely cosmetic. It has no bearing on internal scoring procedures used to determine participant responses to communication signals. Manipulation of the Communication Task, such as the implementation of new Communication audio files, will most likely involve some manual recoding of AF_MATB.

Figure 091 – Comm Slot 2 Label.

Comm Slot 3 Label
This parameter allows the experimenter to change the label of the third communication slot in the communication task.

The default value for this parameter is "COM1" (see Figure 092).

Appropriate values for this parameter are at least one character with a maximum of four or five characters.

PLEASE NOTE: This variable is purely cosmetic. It has no bearing on internal scoring procedures used to determine participant responses to communication signals. Manipulation of the Communication Task, such as the implementation of new Communication audio files, will most likely involve some manual recoding of AF_MATB.

Figure 092 – Comm Slot 3 Label.

Comm Slot 4 Label

This parameter allows the experimenter to change the label of the fourth communication slot in the communication task.

The default value for this parameter is "COM2" (see Figure 093).

Appropriate values for this parameter are at least one character with a maximum of four or five characters.

PLEASE NOTE: This variable is purely cosmetic. It has no bearing on internal scoring procedures used to determine participant responses to communication signals. Manipulation of the Communication Task, such as the implementation of new Communication audio files, will most likely involve some manual recoding of AF_MATB.

Figure 093 – Comm Slot 4 Label.

Slot 1 & 2 Freq. Ranges

Lower Limit

This parameter allows the experimenter to set the minimum frequency for Comm Slots 1 & 2.

The default value for this parameter is 108.7 (see Figure 094).

Appropriate values for this parameter are real numbers greater than zero.

Upper Limit

This parameter allows the experimenter to set the maximum frequency for Comm Slots 1 & 2.

The default value for this parameter is 117.7 (see Figure 094).

Appropriate values for this parameter are real numbers greater than at least one additional increment of the previously mentioned *Lower Limit*.

Figure 094 – Slot 1 & 2 Freq. Ranges.

Slot 1 & 2 Freq. Increments

This parameter defines the increment at which Comm Channels 1 & 2 change.

The default value for this parameter is .2 (see Figure 095).

Appropriate values for this parameter are real numbers greater than zero.

Figure 095 – Slot 1 & 2 Freq. Increments.

PLEASE NOTE: The parameters chosen for the communication frequency range and the increment for that range must be congruent. The example below (Figure 096) demonstrates **Slot 1 & 2 Freq. Ranges** and **Increments** don"t match up. This example will be true for either set of communication slots. A warning will appear (Figure 097) when incongruent parameters are selected.

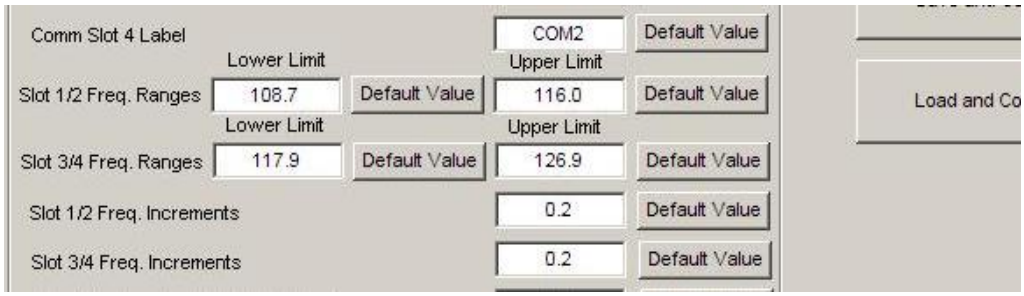


Figure 096 – Example of incongruent frequency limits and increment.

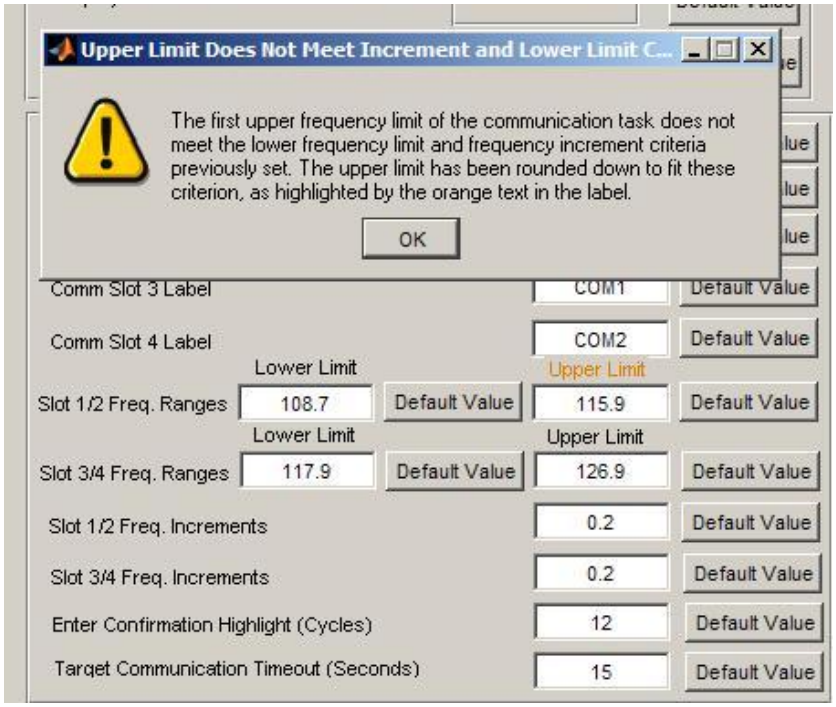


Figure 097 – Warning as a result of attempting to save incongruent frequency limit and increments parameters.

Slot 3 & 4 Freq. Ranges

Lower Limit

This parameter allows the experimenter to set the minimum frequency for Comm Slots 3 & 4.

The default value for this parameter is 117.9 (see Figure 098).

Appropriate values for this parameter are real numbers greater than zero.

Upper Limit

This parameter allows the experimenter to set the maximum frequency for Comm Slots 3 & 4.

The default value for this parameter is 126.9 (see Figure 098).

Appropriate values for this parameter are real numbers greater than at least one additional increment of the previously mentioned *Lower Limit*. For details on increments, see **Slot 3 & 4 Freq. Increments** listed below.

Figure 098 – Slot 3 &4 Freq. Ranges.

Slot 3 & 4 Freq. Increments

This parameter defines the increment at which Comm Channels 3 & 4 change.

The default value for this parameter is .2 (see Figure 099).

Appropriate values for this parameter are real numbers greater than zero.

Figure 099 – Slot 3 & 4 Freq. Increments.

Enter Confirmation Highlight (Cycles)

This parameter defines the number of cycles the Enter button will stay green for to confirm that a channel and frequency were successfully logged. The button will stay green for the specified amount of cycles before returning to its default blue color. This delay is influenced by the speed of the computer that AF_MATB is running on, so if the delay is not long enough due to a fast machine, increase the limit accordingly.

The default value for this parameter is 12 (cycles) (see Figure 100).

Appropriate values for this parameter are integers greater than 0.

Figure 100 – Enter Confirmation Highlight (Cycles).

8.2.5. Tracking Parameters

Tracking Gain

This parameter specifies the sensitivity of the joystick for the Tracking Task.

The default value for this parameter is 65 (see Figure 101).

Appropriate values for this parameter are real numbers greater than 40. At parameters below 25, the joystick may not affect the crosshairs when the speed of the crosshair is at maximum. Therefore, Tracking Gain should greater than 40 to avoid the tracking performance being negatively affected by joystick responsiveness. There is no upper limit for the gain.

Figure 101 – Tracking Gain.

Center Range

Used exclusively for the Tracking‟s Autopilot mode, this pixel value defines how much the crosshair can bounce normally around the center.

The default value for this parameter is 8 (pixels) (see Figure 102).

Appropriate values for this parameter are real numbers greater than 0.

Figure 102 – Center Range.

Autopilot Direction Limit

Used exclusively for the Tracking‟s Autopilot mode, this parameter is designed to determine the maximum amount of time between direction changes for the tracking crosshair. This value is the maximum number of cycles that the crosshair will maintain a certain direction before changing direction. This value is randomly manipulated, so that direction changes are random.

The default value for this parameter is 5 (cycles) (see Figure 103).

Appropriate values for this parameter are real numbers greater than 0. Recommended to be much smaller than the **Manual Direction Limit**.

Figure 103 – Autopilot Direction Limit.

Manual Direction Limit

Used exclusively for the Tracking‟s Manual mode, this parameter is designed to determine the maximum amount of time between direction changes for the tracking crosshair. This value is the maximum number of cycles that the crosshair maintains a certain direction before changing direction. This value is randomly manipulated, so that direction changes are random. This number should be much greater than the **Autopilot Direction Limit**.

The default value for this parameter is 60 (cycles) (see Figure 104).

Appropriate values for this parameter are real numbers greater than 0. However, based on the tracking algorithm, this number should be greater than 10, otherwise the direction changes will be extremely frequent and the crosshair will hover instead of traveling across the screen.

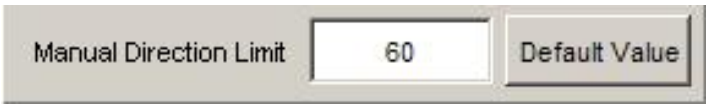


Figure 104 – Manual Direction Limit.

9.0 UTILITY BUTTON DESCRIPTIONS

9.1. Default Value

Next to each parameter is a **Default Value** button. If there are ever any questions about what a particular value should be, you can use this button to give you an example of what the typical parameter‟s value is. This is particularly useful if a manual is not immediately available. This will also restore the parameter label to black, in the case that the label was red or orange due to the input of incorrect values.

9.2. Save Values

This button is used to store a specific set of parameters (see Figure 105). These parameters are stored as a .mat file and can be loaded into AF_MATB_ScriptGenerator, as well as AF_MATB. This button is particularly useful when multiple parameters files need to be saved at one time. A message will be displayed in the GUI that will inform the user that the values were successfully saved.

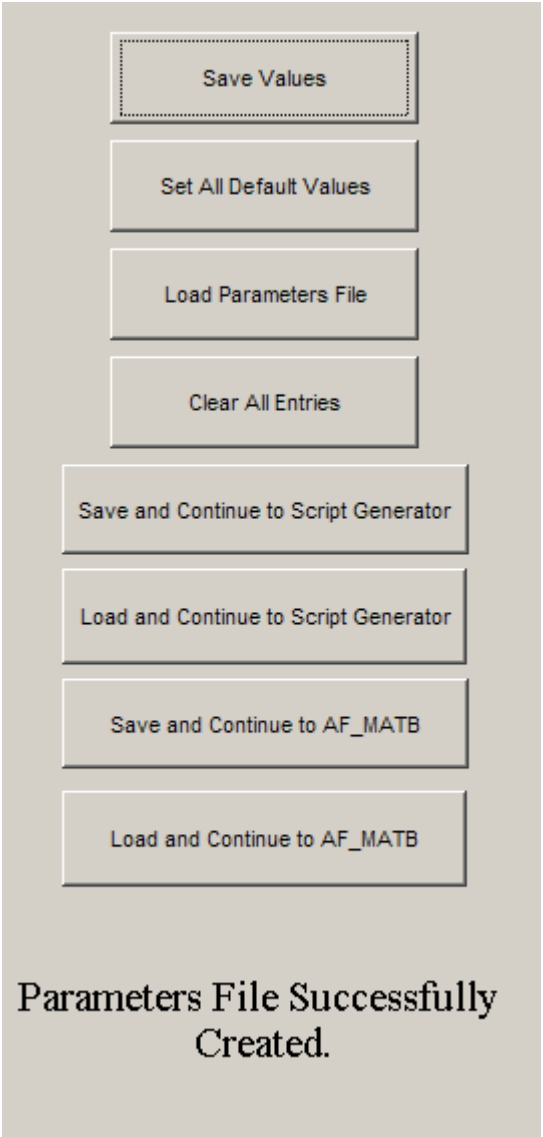


Figure 105 – Button array and information section informing of a successful save.

The save function of this program will automatically verify all parameter values and ensure that they are valid values. Any invalid value will be highlighted in red, and the experimenter will be notified that the parameters were not successfully saved. For example, in the first figure below (Figure 106), **Tank B Starting Volume** contains a value of "Banana," while the second figure (Figure 107) has a missing radio button selection.

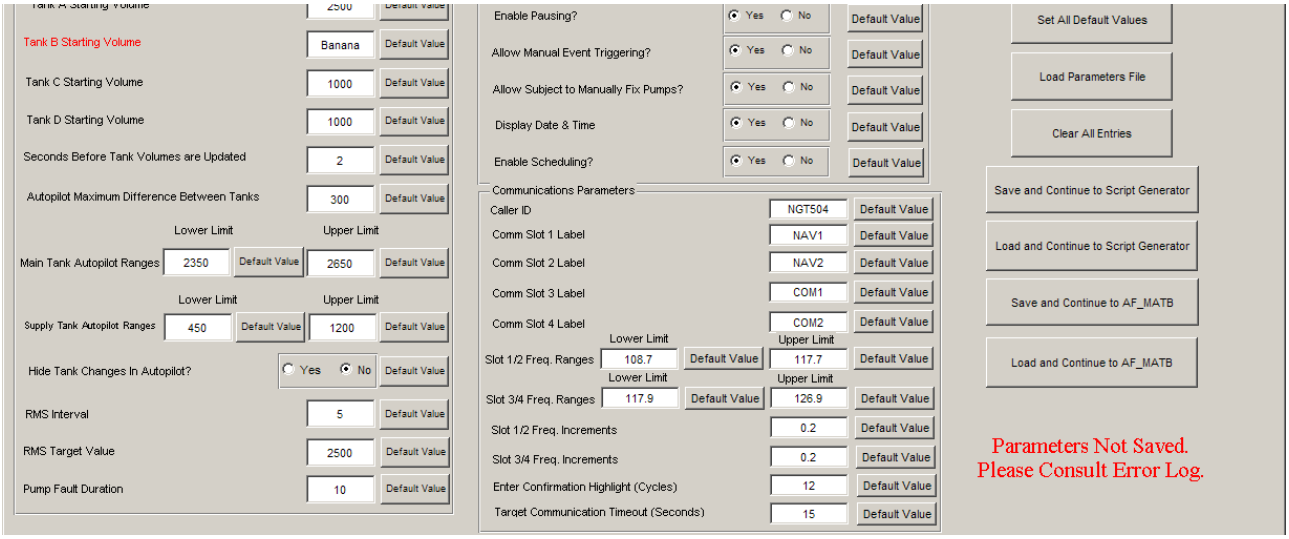


Figure 106 – AF_MATB_Parameters window excerpt illustrating an attempt to save with an invalid parameter entry.

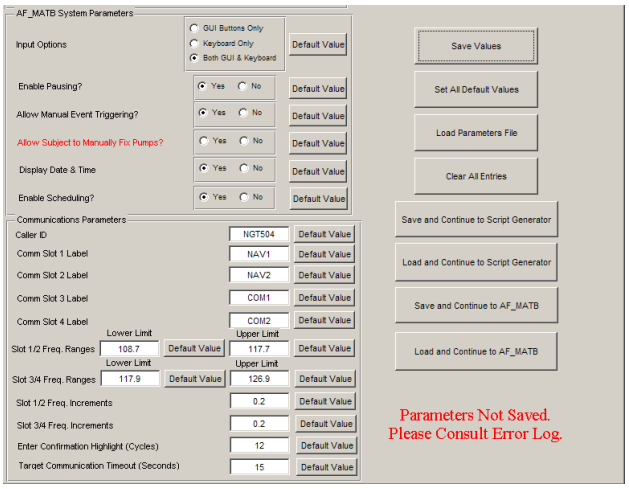


Figure 107 – AF_MATB_Parameters window excerpt illustrating an attempt to save with a missing parameter value.

Suggestions for rectifying invalid parameter values, like those outlined in Figure 108, can be found in the error log. The error log will itemize every error in a parameters file and suggestions for correcting each parameter listed.

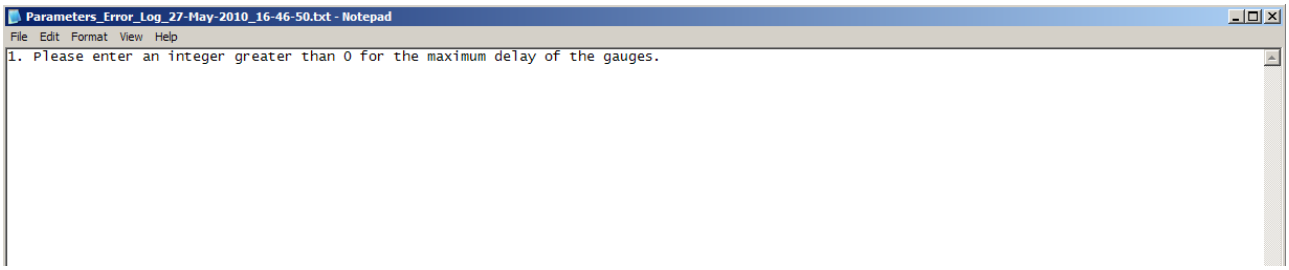


Figure 108 – An example of an error log generated as a result of attempting to save a file with an invalid parameter.

To save a parameters file, click **Save Values**. Once all parameters have been verified as valid, the user would simply select a directory and enter a filename. This filename will automatically have a date and time stamp appended to it to help prevent overwriting. For example, in Figure 109 the user has entered a filename "TestScript1" into the "**File name"** field. A date and time stamp (i.e. 05-Mar-2010_13-50-26) will automatically be appended to the specified filename, as well as a .mat extension. To change the location of the saved script, simply select a new directory from the "**Save in**:" dropdown menu.

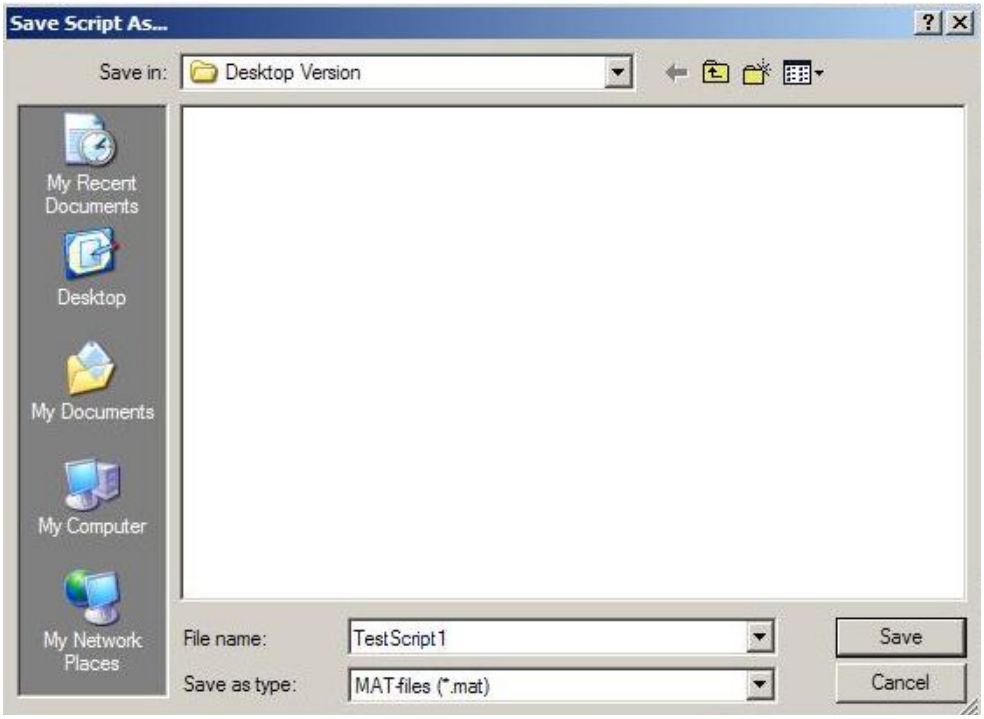


Figure 109 – Standard Windows File Save dialog which the program uses to assist the user in saving parameters files.

9.3. Set All Default Values

This button will restore all parameters to their predefined default values (see Figure 110). Please keep in mind that this will affect ALL parameters. If there are only a limited number of parameters that need to be reset to their default values, it is best to use the **Default Values** buttons next to those specific parameters.

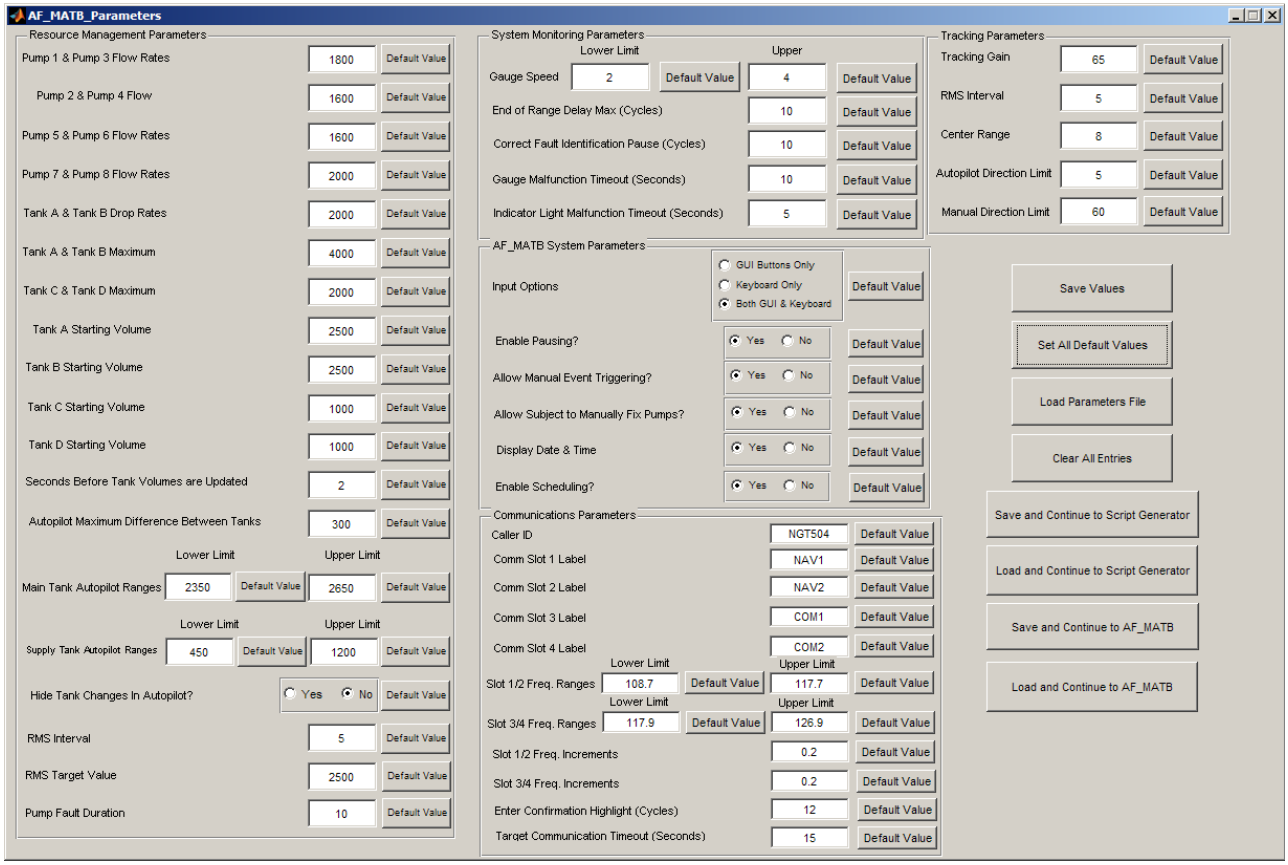


Figure 110 – The AF_MATB_Parameters window after the **Set All Default Values** button was pressed.

9.4. Load Parameters File

This button will load any parameter file (see Figure 111), as well as any successfully constructed script file. The parameters loaded from these files will be automatically updated in the appropriate fields. If loading was successful, a message will be displayed in the GUI, informing you of such.

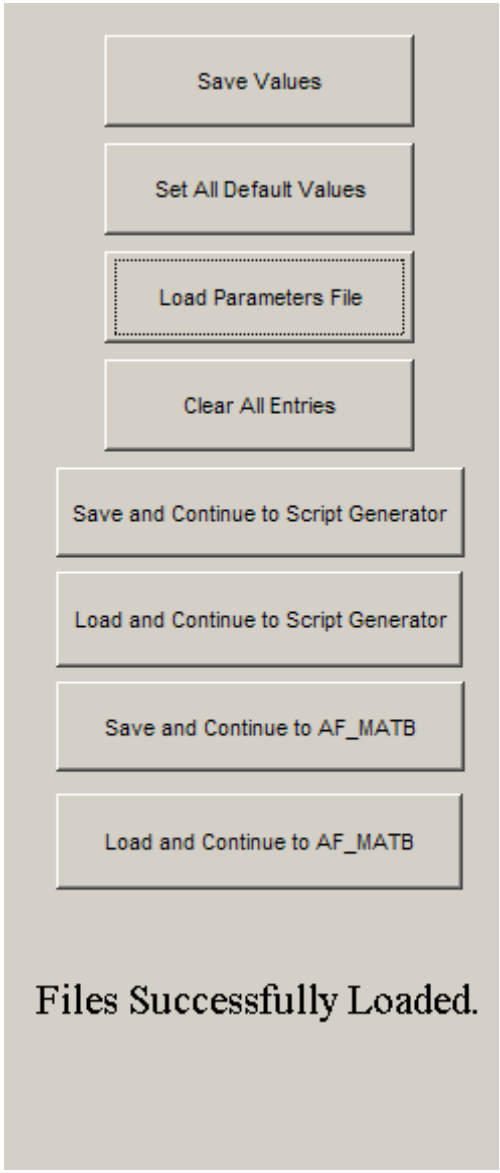


Figure 111 – Button array and information section informing of a successful load.

To load a file, click **Load Parameters File**, and select the directory. Finally, using the file selection dialog see in Figure 112, find and select the proper file.

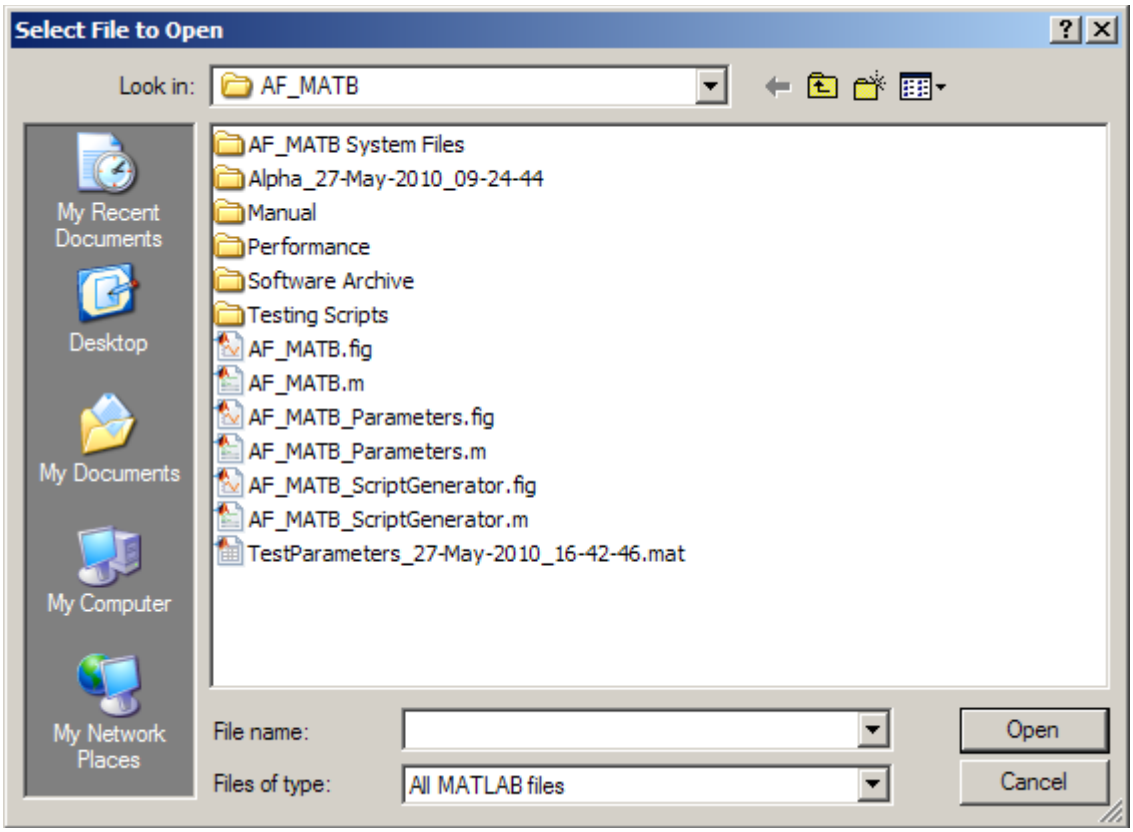


Figure 112 – Standard Windows File Selection dialog which the program uses to assist the user in loading parameters files.

In the event that loading was not successful, the user will be notified (see Figure 113). Either the saving process was interrupted, or the script or parameters file may be corrupted. Users should ensure that all files being loaded are valid scripts compatible with their version of the task.

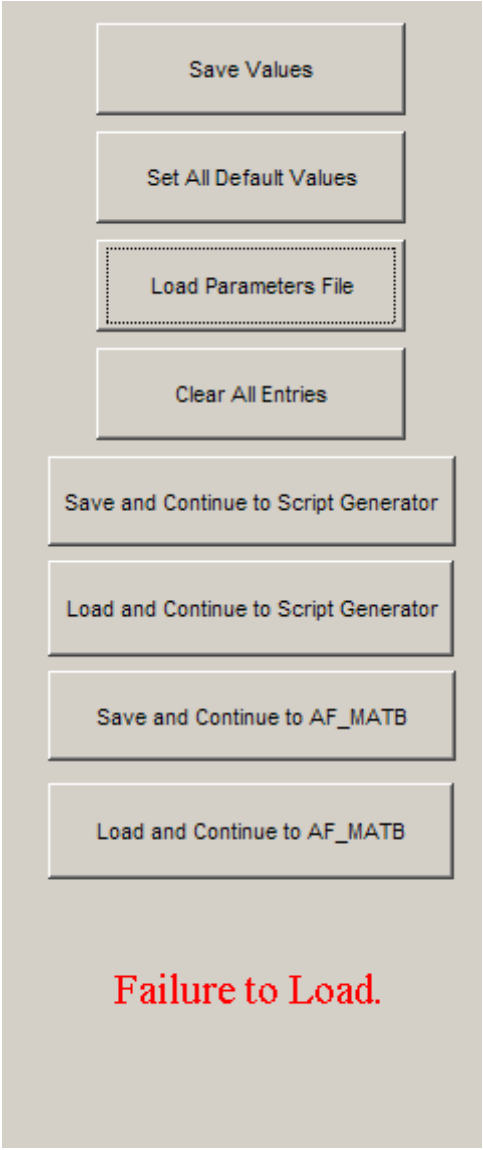


Figure 113 – Button array and information section informing of a failure to load.

9.5. Clear All Entries

This button will clear the entry fields for all parameters in the GUI. The Clear All Entries button is useful when building a custom parameter file (see Figure 114), because it helps to ensure that the user won‟t overlook setting any parameters.

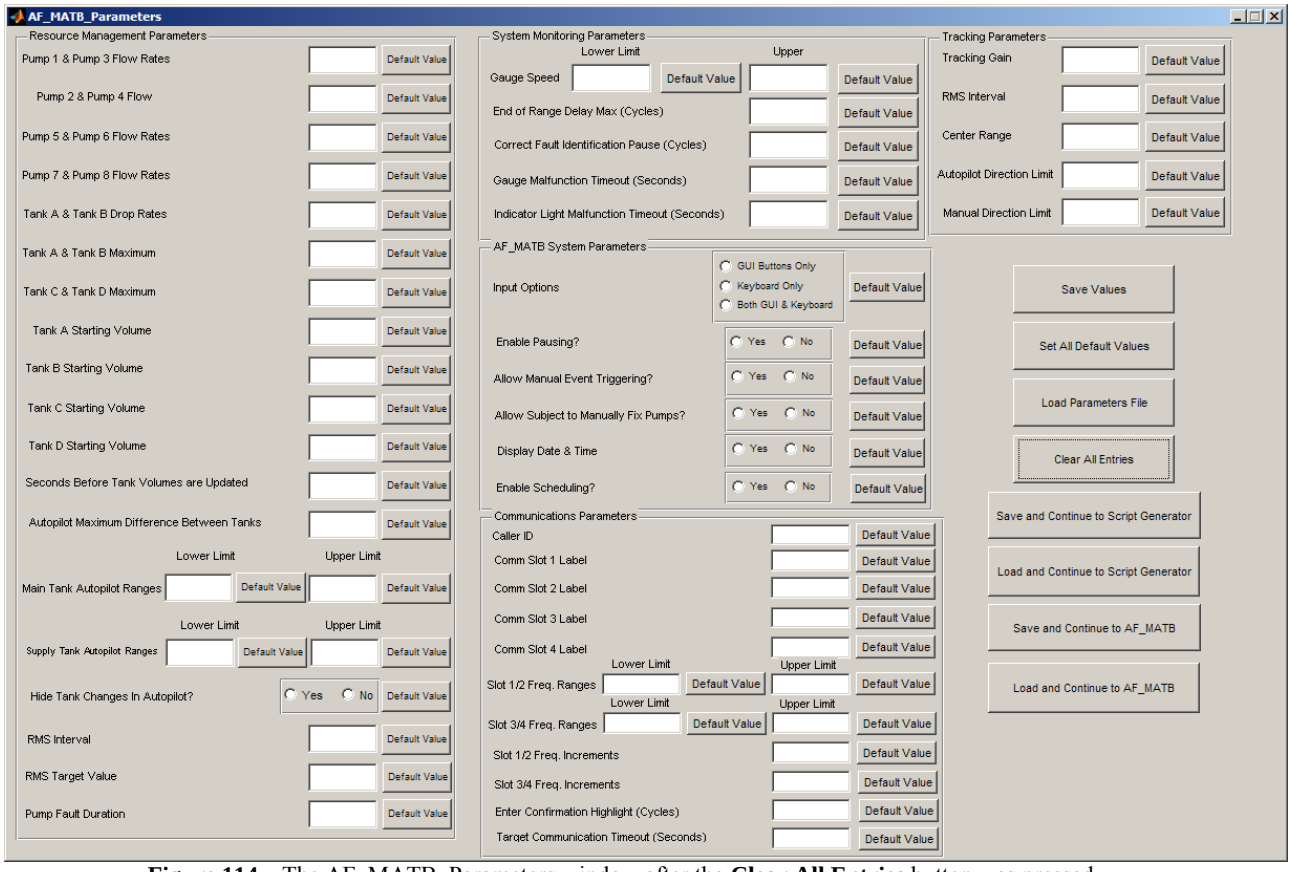


Figure 114 – The AF_MATB_Parameters window after the **Clear All Entries** button was pressed.

9.6. Save and Continue to Script Generator

This button allows the user to save a parameters file and immediately load that list into the Script Generator. This button is useful in situations where a single set of parameters is all that is needed before generating a number of different scripts. If there are multiple sets of parameters to generate, it is recommended that this button is only used for the final script.

To save a parameters file, click **Save and Continue to Script Generator**. Once all parameters have been verified, simply select a directory and enter a filename just as the user would if the **Save Values** button was pressed.

All of the rules that applied to the **Save Values** button apply here as well. After successfully saving a file, the user will be notified that the Script Generator is loading (see Figure 115), and then AF_MATB_Parameters will dispose. Finally, AF_MATB_ScriptGenerator will appear.

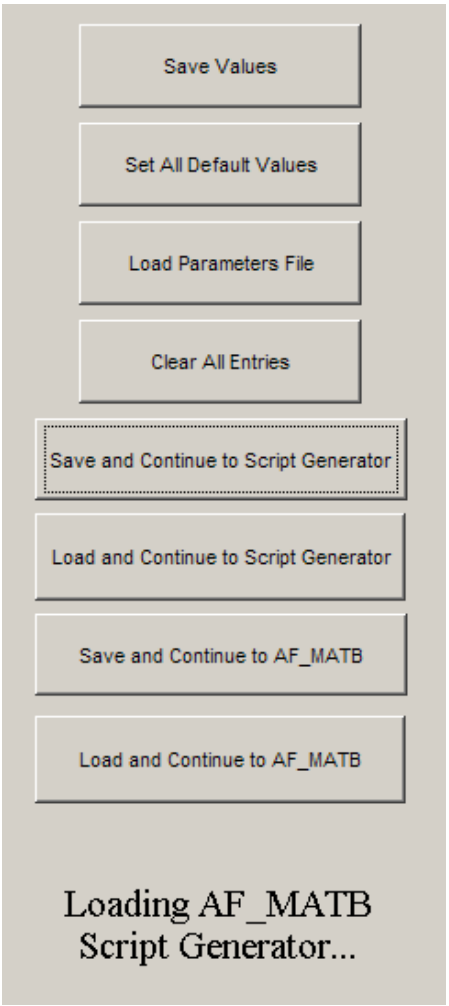


Figure 115 – Button array and information section informing of the loading of AF_MATB_ScriptGenerator after a successful save.

9.7. Load and Continue to Script Generator

This button will load any parameters file, as well as any successfully constructed script file. The parameters loaded from these files will be automatically updated in the appropriate fields and then automatically loaded into the Script Generator. This button would be ideal for individuals who already have a saved parameters file and immediately want to load it into the Script Generator.

To load a file, click **Load and Continue to Script Generator**. Then, select the directory and file using the file selection dialog just as the user would if the **Load Parameters File** button was pressed.

All rules that applied to the **Load Parameters File** button also apply here. If the parameters file loads successfully, a message will be displayed in the GUI to inform the user of this and to notify them that the Script Generator is in the process of loading (see Figure 116). Once it has loaded, AF_MATB_Parameters will dispose and AF_MATB_ScriptGenerator will appear.

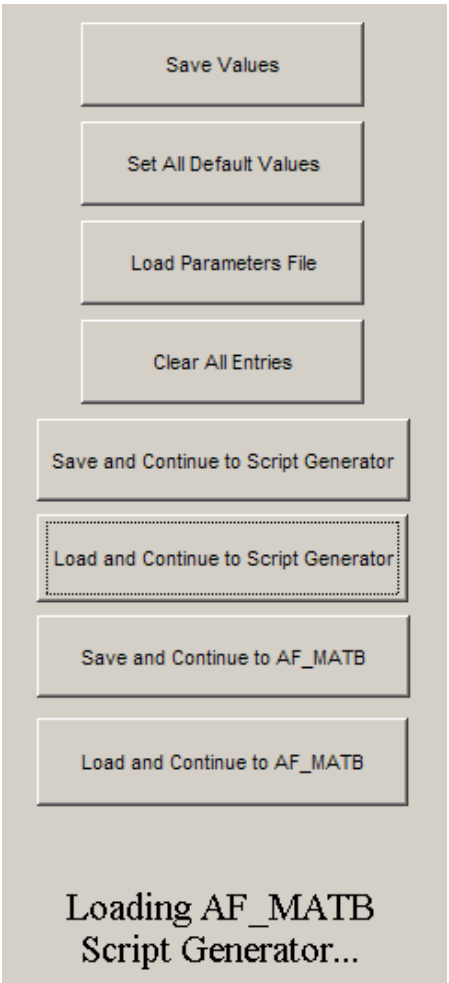


Figure 116 – Button array and information section informing of the loading of AF_MATB_ScriptGenerator after a successful load.

9.8. Save and Continue to AF_MATB

This button allows the user to save a parameter file and immediately load that file into AF_MATB. This button is useful in situations where the experimenter is interested in testing a specific set of parameters, such as pointer speeds, before taking the time to generate a number of scripts.

To save a parameters file, click **Save and Continue to AF_MATB**. Once all parameters have been validated, the user would simply select a directory and enter a filename just as they would if the **Save Values** button was pressed).

All rules that applied to the **Save Values** button also apply here. After successfully saving a file, the user will be notified that AF_MATB is loading (see Figure 117).

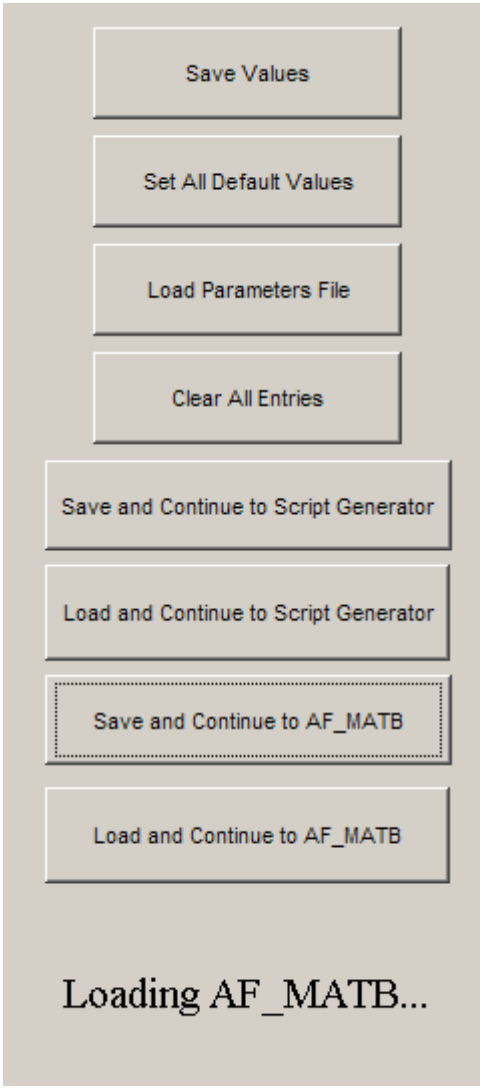


Figure 117 – Button array and information section informing of the loading of AF_MATB after a successful save.

9.9. Load and Continue to AF_MATB

This button will load any parameters file or any successfully constructed script file. The parameters loaded from these files will be automatically updated in the appropriate fields and then loaded into AF_MATB. This button would be ideal for individuals who already have a saved parameters list and immediately want to load it into AF_MATB.

To load a file, click **Load and Continue to AF_MATB**, select the directory and then file using the file selection dialog, just as the user would if the **Load Parameters File** button was pressed.

All rules that applied to the **Load Parameters File** button also apply here. If the parameters file is loaded successfully, a message will be displayed in the GUI informing the user of this (see Figure 118) and to notify them that AF_MATB is in the process of loading. Once it has loaded, AF_MATB_Parameters will dispose, and AF_MATB will appear.

If the parameters do not load successfully, the user should check the script and parameter files; they may be corrupted. This file may be unusable, and the user should restart AF_MATB_ScriptGenerator.

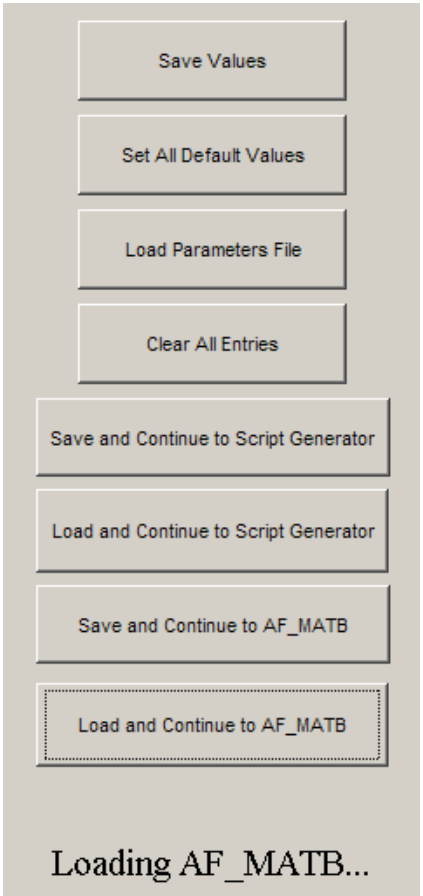


Figure 118 – Button array and information section informing of the loading of AF_MATB after a successful load.

10.0 AF_MATB_SCRIPTGENERATOR "WHY DO I NEED THIS?"

The goal of this utility is to allow for an experimenter to accurately and efficiently create scripts without having to manually manipulate event occurrences. AF_MATB_ScriptGenerator takes into account trial durations, as well as response timeouts for each item in each subtask, to ensure that there will be no overlap between the same two events. For example, this utility ensures that Light 1, while instructed to be turned off from 00:00:05 to 00:00:10, will not receive instructions to be turned off from 00:00:07 to 00:00:12. If this were the case, event durations would be shortened, which would artificially inflate response timeouts. The Script Generator is designed to protect against precisely this, ensuring that performance metrics are not compromised by script instructions. Furthermore, the Script Generator ensures that participants are guaranteed the full allotment of time to respond to a given event. The utility will also disperse events randomly throughout a trial, providing better "random" event times than an individual could manually manipulate on their own. Finally, the utility has built-in protection to maintain the integrity of this event dispersion. Event maximums are set based on trial time and event timeout so that there will be no overlap in response times or event occurrences. This also helps ensure that all events will stay inside a given trial and that their response times will not "bleed" over into the next trial, if applicable.

11.0 SYSTEM REQUIREMENTS

Hardware requirements for the Task Script Generator include 1 GB of RAM, 2 GHz dual-core processor, a keyboard and mouse, and a 15-inch monitor (1280x1024 resolution), at minimum. It is important that these minimum requirements be met to ensure proper function of the utility.

AF_MATB_ScriptGenerator was designed for computers operating on Windows XP SP3 or higher. If the task is executed using a MATLAB script, then version 2007B or later should be used. Otherwise, MATLAB Compiler Runtime 7.8 or later is required.

The Script Generator utilizes Microsoft Excel spreadsheets as a means to output a copy of the script that the experimenter can review. As a result, a copy of Microsoft Excel 2003 or later is required in order for the Script Generator to successfully execute.

Please ensure that your DPI settings are set to default values. If your DPI is set to custom values, you may experience window distortion.

12.0 4 KEY COMPONENTS

The Script Generator can be broken down into 4 key parameters. The first set of parameters is the event timeouts, set in the AF_MATB_Parameters utility. The Script Generator directly utilizes those parameters in determining maximum event rates and will code events while taking this into account. Please see AF_MATB_Parameters‟ *Script-Critical Parameters* section for more information about redefining these parameters.

The second parameter critical to the proper functioning of the Script Generator is a valid trial length. Trial lengths can be entered as either seconds or minutes. What defines a valid trial length will be discussed later in this section.

The third set of critical parameters in this utility is valid event rates. This utility simply takes a specific number of event occurrences in a given timeframe, and shuffles each of them around so that they meet all the necessary requirements of a proper script. More detail on the algorithms used in this utility will be discussed later.

The fourth and final parameter for this utility is the trial condition. The Script Generator will not generate a script until a trial condition has been established in the “List of Conditions” list box. The minimum number of trials necessary to create a script is one, and there is no maximum. This allows the experimenter to create scripts that are of considerable length and variety, which allows for more freedom in the experimental design.

13.0 "HOW DOES THIS UTILITY WORK?"

The Script Generator uses the 4 critical parameters described in the previous section to first determine if the maximum number of events per trial has been exceeded. If this is the case, or if the user has input any invalid parameters, the user will be notified through the error log, precisely like the one used in AF_MATB_Parameters. This log will have a date and time stamp appended to it, and outline precisely what parameter requirements have not been met, with recommendations on what acceptable values may be used.

Once all input parameters have been validated, the algorithm divides up all subtasks into channels. For example, in the System Monitoring Subtask, there are 6 event channels, four channels for each of the four gauges, and 2 channels for each of the two lights. The event occurrences are then distributed across all channels. In the case of odd-numbered events, such as 13, the event occurrences are distributed in an orderly fashion, so that all channels have 3 events and 1 channel will have an additional event, etc. The only other subtask that has multiple channels is the Resource Management subtask, where each of the eight pumps is a separate channel.

Once the event occurrences have been divided into channels, they are distributed across the specific trial length where it is verified that they fit in the trial and do not exhibit any response overlap. Once all channel timelines have been constructed, the channel timelines are merged back together into a subtask timeline, and eventually, into a master timeline.

This master timeline is arranged in a serial fashion, with the earliest event going first and the latest event going last. The timeline is not segregated by subtask or channel, so events from one subtask will be directly next to events from a different subtask. AF_MATB will read this timeline in a serial fashion and trigger each event as soon as its onset time occurs.

It is important to mention a few key interactions between the script and the task architecture. AF_MATB is designed to cycle every .1 seconds and does so reliably on adequately equipped computers (please see AF_MATB"s hardware requirements for details on adequately equipped computers). Even so, events that are all scheduled to occur inside this window of time can still be executed without problem. For example, in a situation where an event occurs at .12 seconds and another at .17 seconds, both events will be triggered when the task cycles next at .2 seconds. Due to the programming architecture of MATLAB, this was the most efficient and reliable way to trigger events without putting further constraints on event times. Please note that the delayed onset will not compromise response times, as they are adjusted based on the difference between the scheduled onset and true onset.

14.0 SCRIPT GENERATOR OUTPUT

After a script has been successfully generated, 2 files will be created. One is a .mat file, which is the file that will be loaded into AF_MATB. The other file is an Excel workbook comprised of 2 worksheets. The first worksheet displays a timeline that is identical to the .mat file that will be read by the task. Each column in the first worksheet conveys certain information.

Column 1 details the time the event is proposed to occur, while Column 5 proposes the timeout for that event. Please keep in mind that these numbers are theoretical and may vary slightly depending on task functioning. Column 2 is used to identify each subtask and channel, while Column 3 defines a particular channel, only. Finally, Column 4 details the exact event that is proposed to occur.

Each event has a unique code, with no code being duplicated. Worksheet 2 is a key that details what actions correspond to which numbers in Column 4. Please note that Worksheet 2 itemizes event codes for all of AF_MATB, not just those event codes the Script Generator uses. As such, there will be a number of event codes that the user will never see encoded in the actual script file.

Both files will be named with whatever filename is specified by the user. In addition, the date and time stamp will also be appended to the files to protect the data from being overwritten.

For an example Excel script, please see Appendix C.

15.0 SCRIPT GENERATOR PARAMETERS

15.1. Sequence Length

Seconds
This parameter defines the length of time for each trial in seconds. Entering values for this parameter will automatically update the **Minute** counterpart. This parameter is considered the primary parameter for determining length.

Default values for each trial are 300, 300, 300, and 60 for the Low, Moderate, High, and Transition trials, respectively.

Appropriate values for this parameter are integers greater than zero, though due to the nature of the task, it would not make sense to set trial lengths less than 10 seconds.

Minutes
This parameter defines the length of time for each trial in minutes (see Figure 119). Entering values for this parameter will automatically update the **Second** counterpart. This parameter is considered the primary parameter for determining length.

Default values for each trial are 5, 5, 5, and 1 for the Low, Moderate, High, and Transition trials, respectively.

Appropriate values for this parameter are real numbers greater than zero, though due to the nature of the task, it would not make sense to set trial lengths less than .17 minutes (10 seconds).

Condition	Sequence Length	
	Seconds	Minutes
Low Difficulty	300	5
Moderate Difficulty	300	5
High Difficulty	300	5
Transition Period	60	1

Figure 119 – The **Sequence Length** section of the Script Generator"s containing the default values. **Condition** column added for reference.

15.2. Communications

Target Callsign
This parameter defines the number of communications that will be directed to the participant in a given trial length. These are messages that the participant is instructed to respond to and should correspond with the **Caller ID** parameter set in AF_MATB_Parameters.

Default values for each trial are 7, 12, 16, and 1 for the Low, Moderate, High, and Transition trials, respectively.

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is dependent on the amount of time scripted for timeouts, as well as the duration of the trial and the length of each communication. (By default, AF_MATB"s communications are approx. 8 seconds in length.)

Distractor Callsign
This parameter defines the number of communications designated to the another plane in a given trial length. These are messages that the participant is instructed to ignore.

Default values for each trial are 1, 4, 6, and 0 for the Low, Moderate, High, and Transition trials, respectively (see Figure 120).

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is dependent on the timeout, as well as the duration of the trial and of each communication.

PLEASE NOTE: Maximum values for **Target** and **Distractor** communications are determined by summing the 2 event counts together. For example, the maximum number of communications for the Communications Task, given default parameters, is 23. Thus, between the **Target** and **Distractor** communications, there can only be 23 events, though they can be split in any way. The Script Generator will only notify the user if the sum is greater than the calculated maximum.

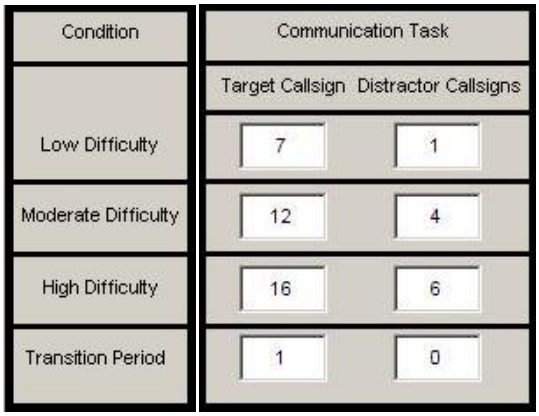


Figure 120 – The **Communication Task** section of the Script Generator"s parameters containing default values. **Condition** column added for reference.

15.3. Tracking

Difficulty
This parameter defines the inputs for the tracking algorithm which manipulate the task‟s difficulty. There are three levels of difficulty: Low, Moderate, and High. There is also the option to utilize the Tracking Autopilot. In the Transition trials, an exclusive option was added to manipulate tracking difficulties before or after the transitions. This option is called “Start Next Stage.” It triggers the difficulty of the tracking task during the transition independent of the other tasks in the battery. Thus, the tracking task can switch levels of difficulty before the other tasks, or it can switch to that level of difficulty after the other tasks in the battery have switched. The option to “Start Next Stage” was deemed to be useful for workload studies run in the past.

Default values for each trial are “TRACKING LOW,” “TRACKING MODERATE,” “TRACKING HIGH,” and “Start Next Stage” for the Low, Moderate, High, and Transition trials, respectively.

Appropriate values for this parameter are: “AUTOPILOT,” “TRACKING LOW,” “TRACKING MODERATE,” and “TRACKING HIGH”. For Transition trials, the “Start Next Stage” option was also added.

Seconds
This parameter, in conjunction with “Start Next Stage,” defines how many seconds before or after a trial ends that the new tracking difficulty should be triggered. This parameter is exclusive only to Transition trials and is only enabled when the tracking difficulty for that trial is selected as “Start Next Stage.”

Default values only exist for the Transition trial, with a default value of 10.

Appropriate values for this parameter are real numbers greater than 0, but less than the length of the Transition trial.

Before/After Next stage

During a Transition trial, the tracking task can be manipulated independent from the other tasks in the battery. At the beginning of a Transition trial, the tracking task can be coded to switch to the new level of difficulty (the difficulty set for the transition) before the other tasks switch. This is the „Before" option. Alternately, the tracking task can switch to the new level of difficulty level for the transition trial after the other tasks have switched. This is the „After" option. This parameter is exclusive only to Transition trials and is only enabled when the tracking difficulty for this trial is selected as "Start Next Stage."

Default values only exist for the Transition trial, with a default value of "BEFORE" (see Figure 121).

Appropriate values for this parameter are "BEFORE" or "AFTER"

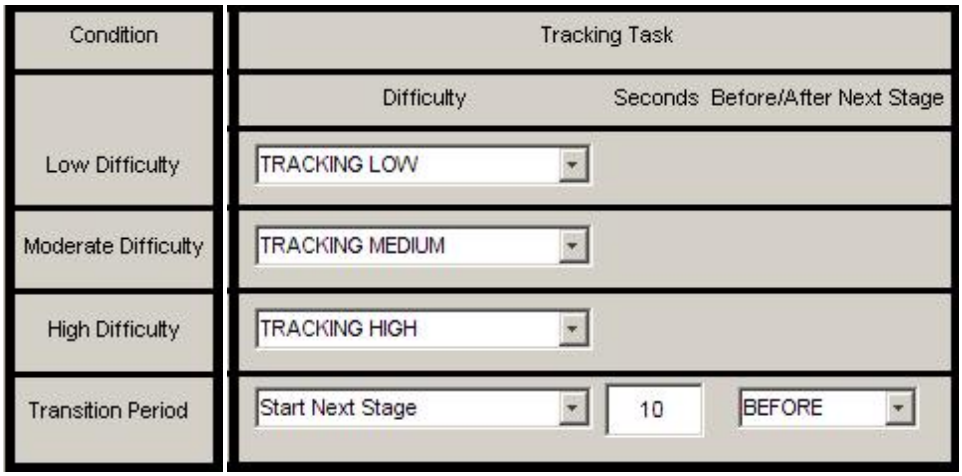


Figure 121 – The **Tracking Task** section of the Script Generator"s parameters containing default values. **Condition** column added for reference.

15.4. System Monitoring

Lights
This parameter defines the number of light faults that will occur per trial. This parameter is defined as the total number of events for the System Management‟s Lights subtask, not for each individual channel. For more information on subtask channels, see the "**How does this utility work?"** section discussed earlier in this chapter.

Default values for each trial are 10, 25, 60, and 2 for the Low, Moderate, High, and Transition trials, respectively (see Figure 122).

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is dependent on the length of time specified for timeouts, as well as the duration of the trial.

Gauges
This parameter defines the number of gauge faults that will occur per trial. This parameter is defined as the total number of events for the System Management‟s Gauges subtask, not for each individual channel. For more information on subtask channels, see the "**How does this utility work?"** section discussed earlier in this chapter.

Default values for each trial are 11, 26, 60, and 2 for the Low, Moderate, High, and Transition trials, respectively (see Figure 122).

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is dependent on the length of time specified for timeouts, as well as the duration of the trial.

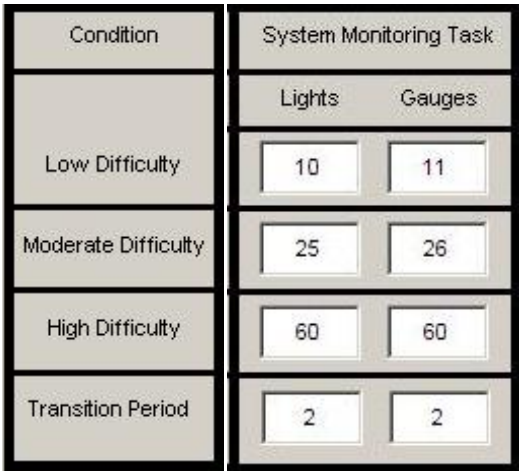


Figure 122 – The **System Monitoring Task** section of the Script Generator‟s parameters containing default values. **Condition** column added for reference.

15.5. Resource Management

Failures & Fixes
This parameter defines the number of Resource Management pump failures that will occur during a given trial. This parameter is defined as the total number of events for this fault-type in the Resource Management subtask, not the number of events per channel. For more information on subtask channels, see the "**How does this utility work?"** section discussed earlier in this chapter. For more information on Resource Management pump failures, see the **Resource Management s**ection in the AF_MATB chapter.

Default values for each trial are 2, 5, 11, and 0 for the Low, Moderate, High, and Transition trials, respectively (see Figure 123).

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is dependent on the length of time specified for timeouts, as well as the duration of the trial.

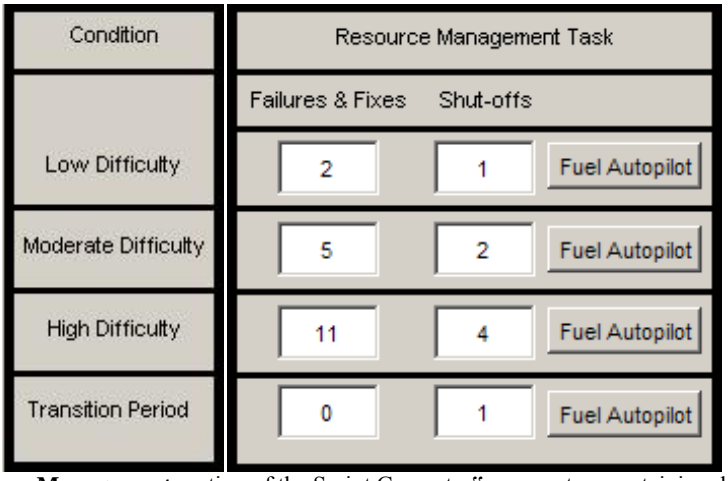


Figure 123 – The **Resource Management** section of the Script Generator"s parameters containing default values. **Condition** column added for reference.

Shut-offs
This parameter defines the number of Resource Management pump shut-offs during a given trial. This parameter is defined as the total number of events for this fault-type in the Resource Management subtask, not the number of events per channel. For more information on subtask channels, see the "**How does this utility work?"** section discussed earlier in this chapter. For more information on Resource Management pump shut-offs, see the **Resource Management s**ection in the AF_MATB chapter.

Default values for each trial are 1, 2, 4, and 1 for the Low, Moderate, High, and Transition trials, respectively (see Figure 124).

Appropriate values for this parameter are integers greater than or equal to 0. The maximum value is recommended to be 50% of the **Failures & Fixes** value for the corresponding trial. This is because a pump shut-off will automatically override a pump failure, so to ensure that most, if not all, failures last for their full duration, the number of shut-offs must be less. If this number exceeds 50% of the **Failures & Fixes** value, the user will be notified of this by a warning message (Figure 124). However, despite the warning, the script can still be generated even if the parameters set do not meet the recommended values.

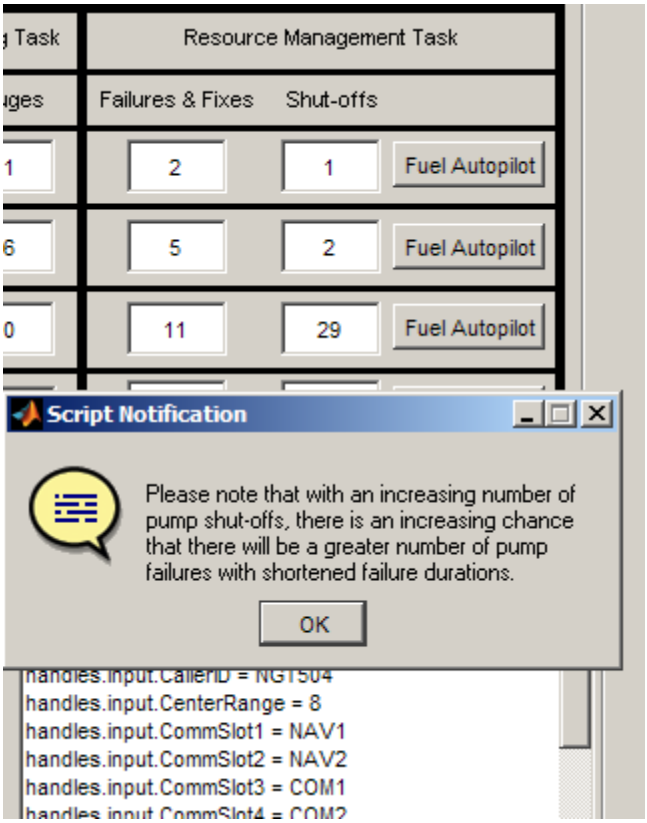


Figure 124 – Pump Shut-off notification if the number of **Shut-offs** exceeds 50% of the number of **Failures & Fixes**.

Fuel Autopilot
This button in the Resource Management section of the Script Generator allows the user to toggle between setting Failure and Shut-off parameters or enabling the Fuel Autopilot for a given trial.

To enable the Fuel Autopilot, simply click the **Fuel Autopilot** button. A dropdown menu will appear, with "AUTOPILOT ENABLED" as the current selection (see Figure 125).

To disable the autopilot and return to the screen with the Failure/Shut-off parameters, simply click the dropdown menu, and select "Enter Task Numbers…" This will cause the dropdown menu to disappear and the original input parameter boxes to reappear. At this point, the Fuel Autopilot is now disabled.

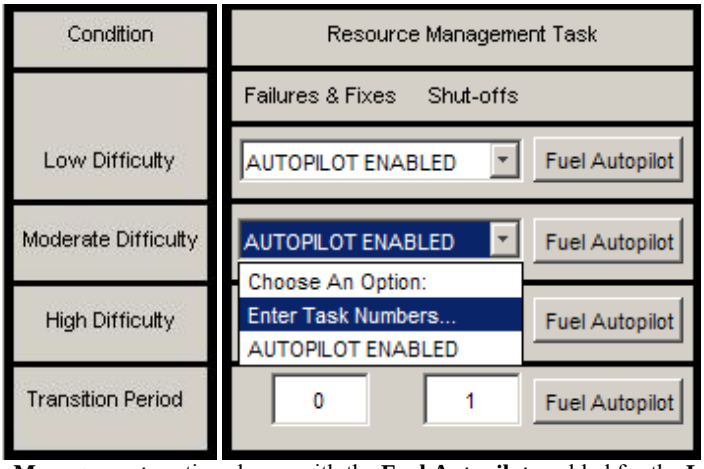


Figure 125 – The **Resource Management** section shown with the **Fuel Autopilot** enabled for the **Low Difficulty** trials and the **Moderate Difficulty** trial being configured from **Fuel Autopilot** back to the standard **Failures & Fixes** and **Shut-offs** parameters.

16.0 CONDITION LIST

The Condition List is designed to indicate the trial structure of a given script. This list allows you to see the arrangement of trials throughout a given run, as well as delete trials from the list. To select a given trial, simply click the **Select Low**, **Moderate**, or **High Difficulty** buttons to the right of the Condition List. The corresponding difficulty will then appear.

To enable transitions between trials, select the "Yes" radio button from the **Enable Transitions?** box. From this point on, between any two new trials that are added to the Condition List, a "Transition Stage" will be inserted. Please note that enabling transitions is not retroactive; therefore, if the user has trial difficulties already entered in the list, and would like to have transitions enabled, they would need to delete those conditions from the Condition List, enable transitions, and then reselect those trials (see Figure 126).

To delete trials, click on the trial in the Condition List and then click the **Delete Condition** button to the right of the Condition List. Any trial, including Transition Stages, can be deleted.

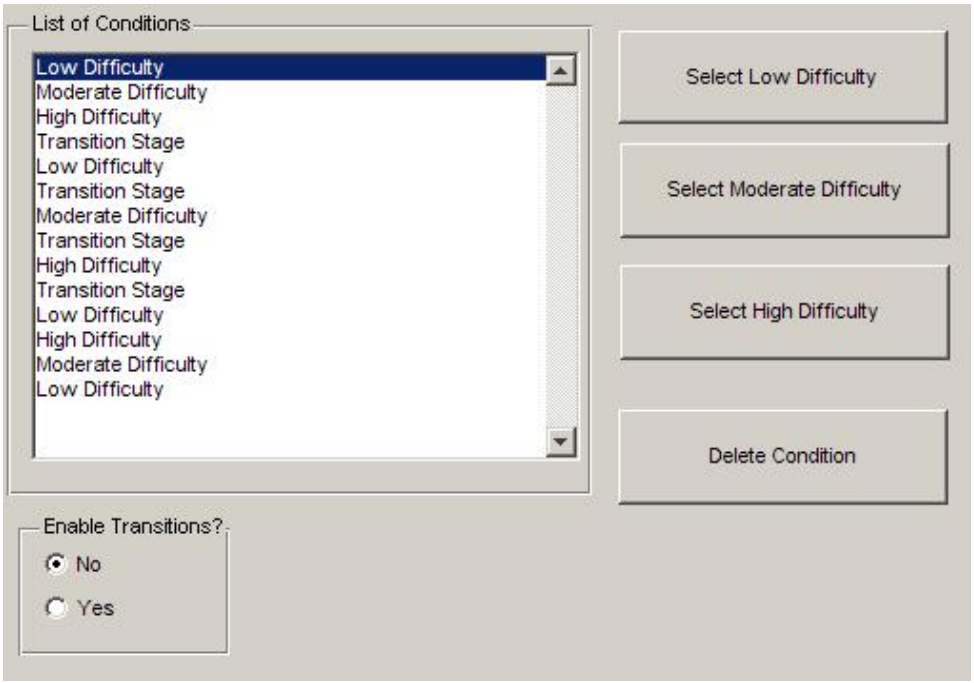


Figure 126 – The Script Generator"s **List of Conditions** section, which details how many trials and of what difficulties the trials in a particular script will be.

17.0 VARIABLES CURRENTLY LOADED

This list indicates the values of all AF_MATB_Parameters variables that were transferred into the Script Generator, or the default values that were designated by the Script Generator. This list is particularly useful when the experimenter would like to verify what parameters the Script Generator will use (see Figure 127).

To erase parameters loaded into AF_MATB_ScriptGenerator, click the **Delete All Variables** button. To restore parameters to their default values, click the **Set Variable Defaults** button.

Please be aware that if the user clears the Variables Currently Loaded List and then attempts to generate a script, the Script Generator will not function as expected. The user should ensure that either the default values, or custom parameters are loaded into the Script Generator and that those variables appear in this list.

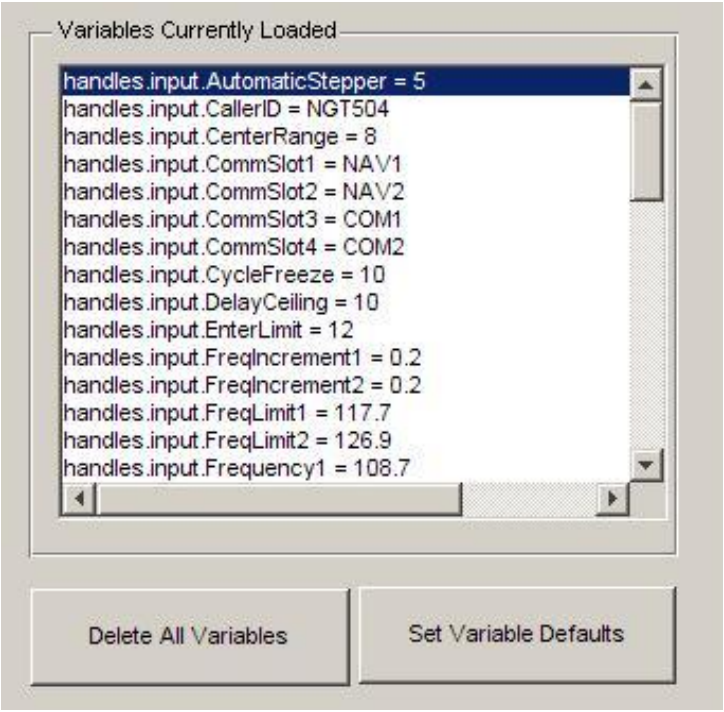


Figure 127 – The **Variables Currently Loaded** section, which details precisely what parameters the Script Generator will use when constructing and saving a script.

18.0 UTILITY BUTTON DESCRIPTIONS

18.1. Generate Script

This button is used to generate a single script to be loaded into AF_MATB at a later time.

The save functionality of this program is very similar to that of AF_MATB_Parameters in that it will automatically verify all parameter values and ensure that they are valid values. Any invalid value will have a red parameter label, and the experimenter will be notified that script was not saved (Figure 128).

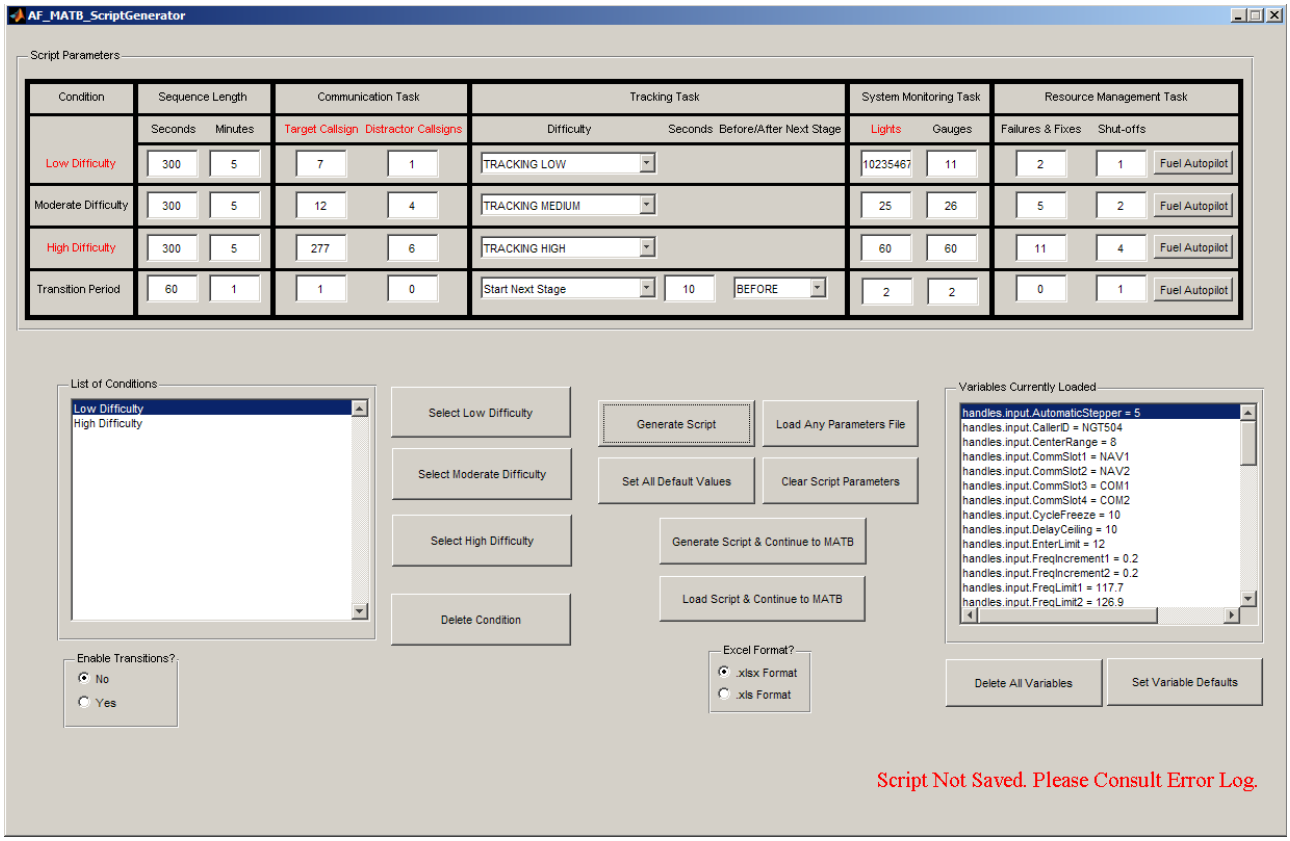


Figure 128 – AF_MATB_ScriptGenerator after someone has attempted to generate a script using invalid script parameters.

Suggestions for rectifying invalid parameter values can be found in the error log (Figure 129).

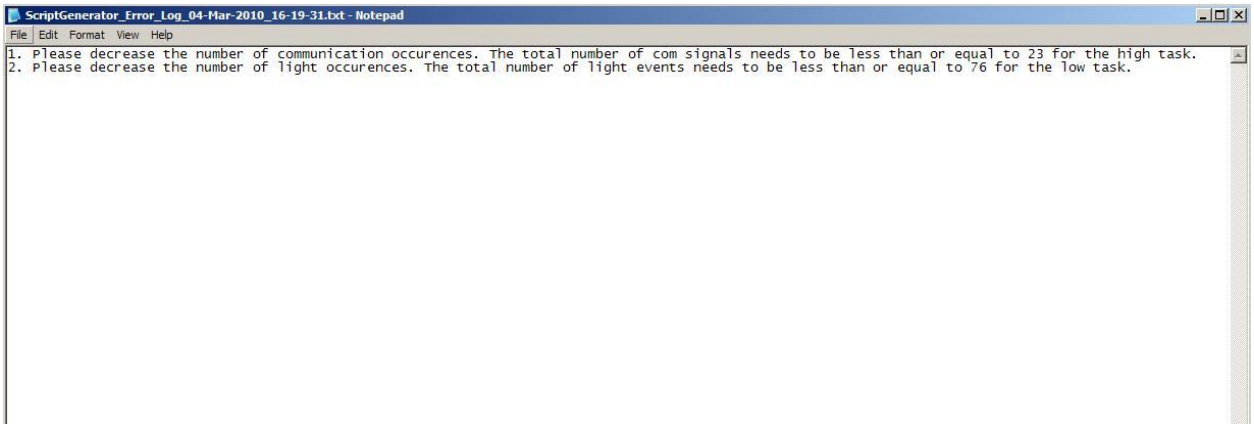


Figure 129 – An error log generated by AF_MATB_ScriptGenerator in response to the invalid parameters used in **Figure 128**

To save a script, click **Generate Script**. Once all parameters have been verified, simply select a directory and enter a filename. This filename will automatically have a date and time stamp appended to it to prevent overwriting.

A message will then appear in the GUI, informing the user that the script Excel file is currently being constructed (Figure 130). Please wait, as script construction may take some time, especially if multiple trials or long trial lengths are included in a script.

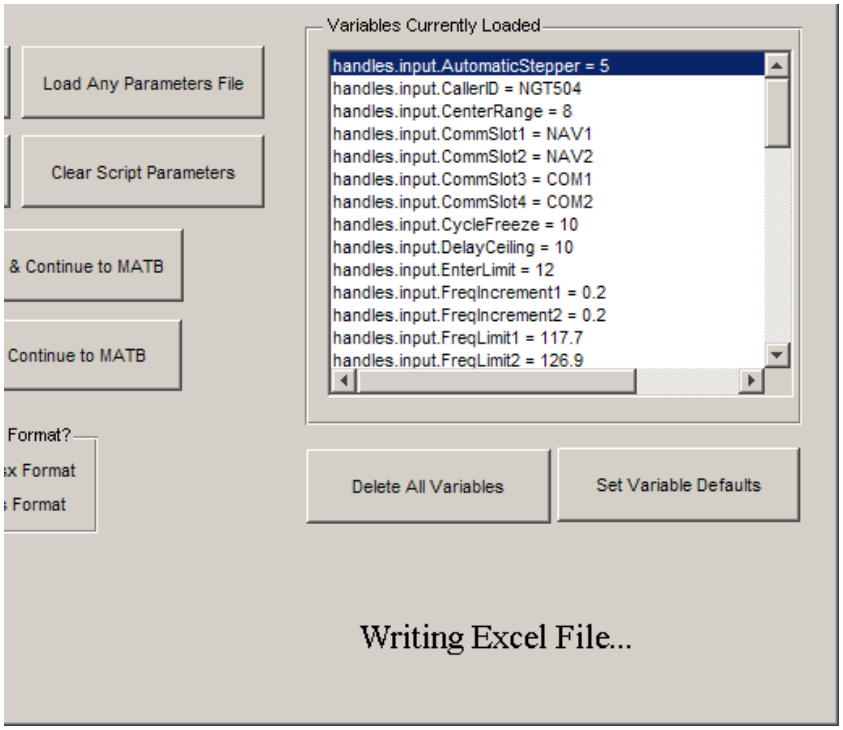


Figure 130 – The information section of the window, informing the user that script generation is in progress.

When both parts of the script file have been successfully constructed, the task window will once again become functional, and the user will be notified of the successful script generation (see Figure 131).

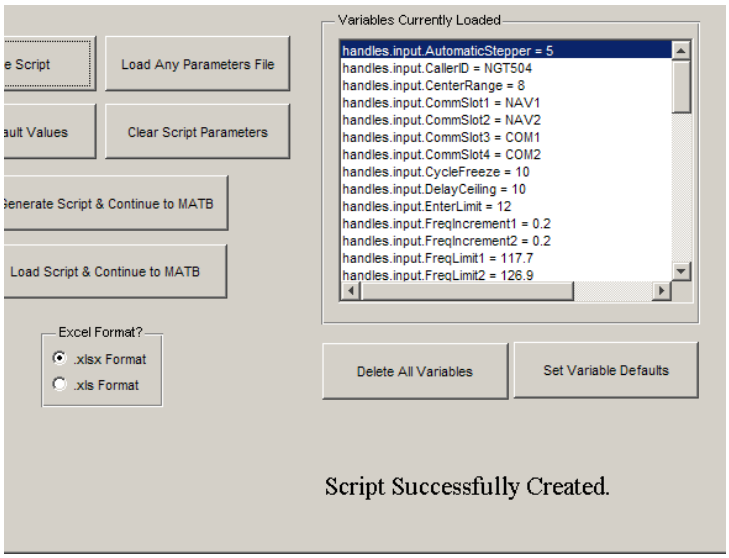


Figure 131 – Information section of the window, informing the user of a successful script generation.

In the event that no trials were selected by the user, or that there was an unforeseen error by the Script Generator, the following message will appear (see Figure 132).

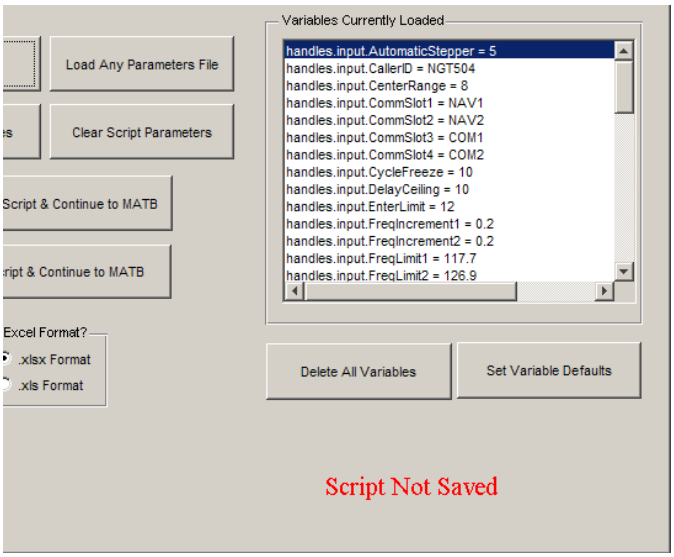


Figure 132 – Information section of the window, informing the user of a failure to save due to no trials being entered into the Script Generator or another unforeseen generation error.

PLEASE NOTE: As previously stated, each script that is generated outputs 2 files, a .mat file and a Microsoft Excel file. Since some computers may not have upgraded to the new Microsoft Office 2007 format or do not have the compatibility pack, the option was added to allow for a choice in file format. At the time that a script is generated, whatever radio button is selected in the **Excel Format?** box (Figure 133) is what that script file‟s Excel counterpart will write as.

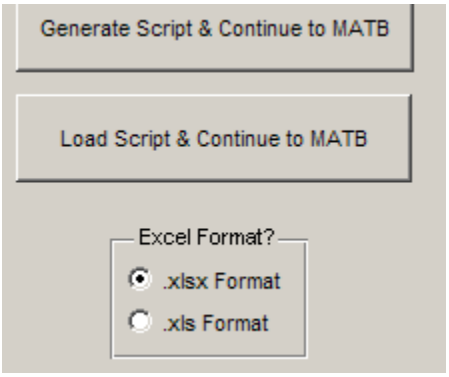


Figure 133 – The **Excel Format?** section, which allows the user to toggle between Excel 2003 and Excel 2007 formats for convenience.

For an example Excel script, please see Appendix C.

18.2. Load Any Parameters File

This button will load any parameters file, as well as any successfully constructed script file. The parameters loaded from these files will be automatically updated in the appropriate fields. If the parameters were successfully loaded, a message will be displayed in the GUI, informing the user of such. Furthermore, the Script Generator is able to distinguish between the loading of a script file and the loading of a parameters file. In Figure 134, the user is notified that both script parameters, as well as task parameters, were loaded into the Script Generator.

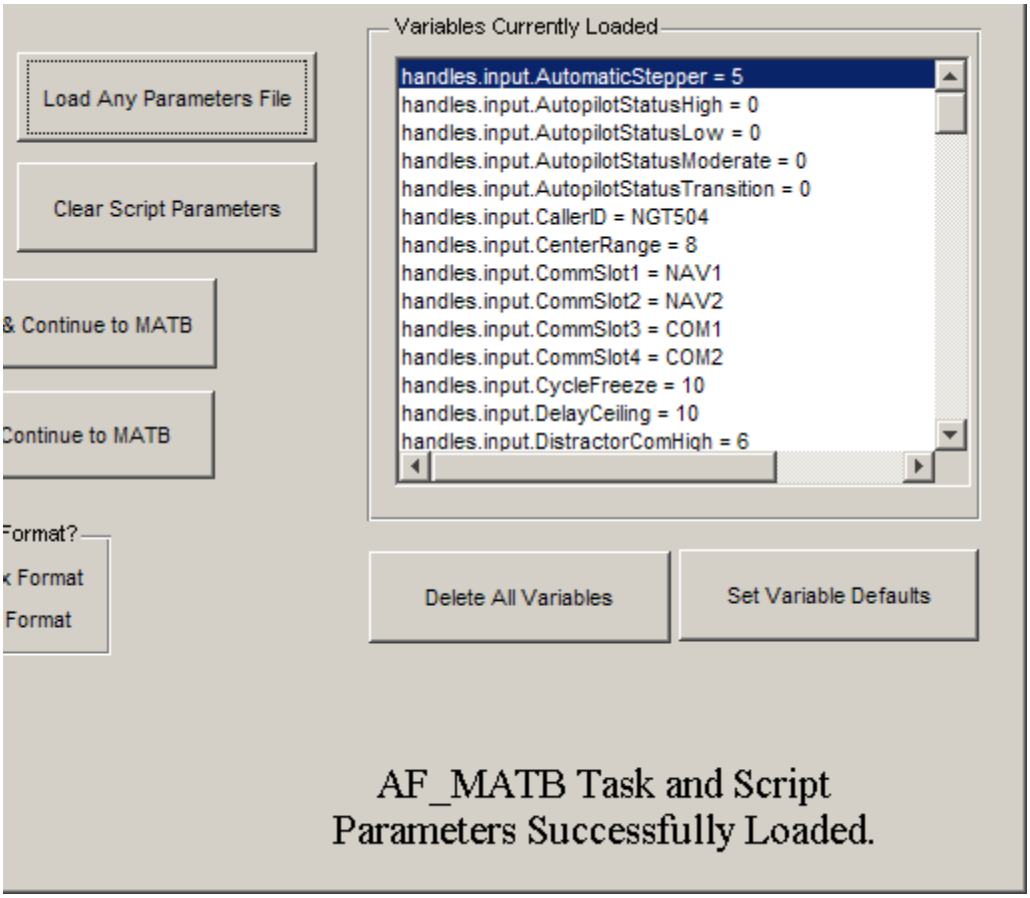


Figure 134 – Information section, informing the user that both task parameters and script parameters were loaded from the selected file.

In this example (Figure 135), however, the user is notified that only a custom parameter file was loaded into the script.

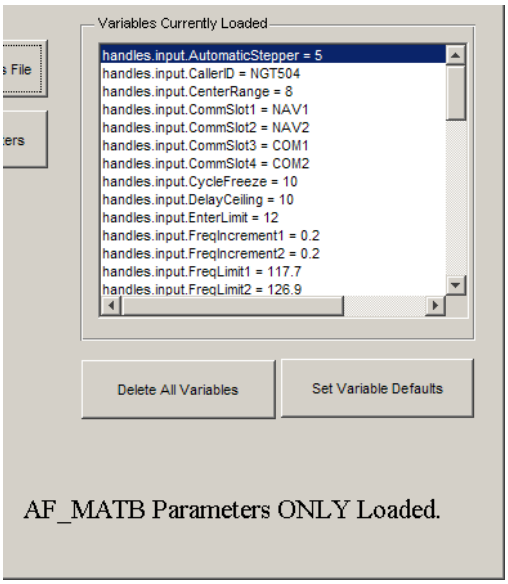


Figure 135 – Information section, informing the user that a task parameters file was selected, and as a result, only task parameters were loaded from that file.

To load a file, click **Load Any Parameters File**, select the directory, and then select the file using the file selection dialog.

If the user finds that the file will not successfully load (Figure 136), it is possible that the script or the parameters file is corrupt, and the file may be unusable at this point.

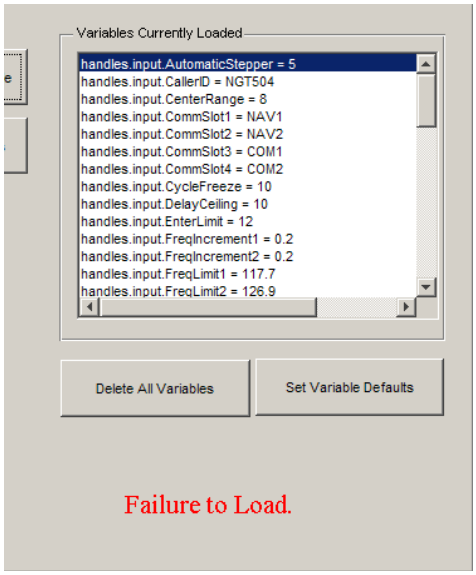


Figure 136 – Information section, informing the user of a failure to load due to a corrupt script or other unforeseen loading error.

18.3. Set All Default Values

This button will restore all script parameters to their predefined default values. Please keep in mind that this will affect ALL parameters (see Figure 137).

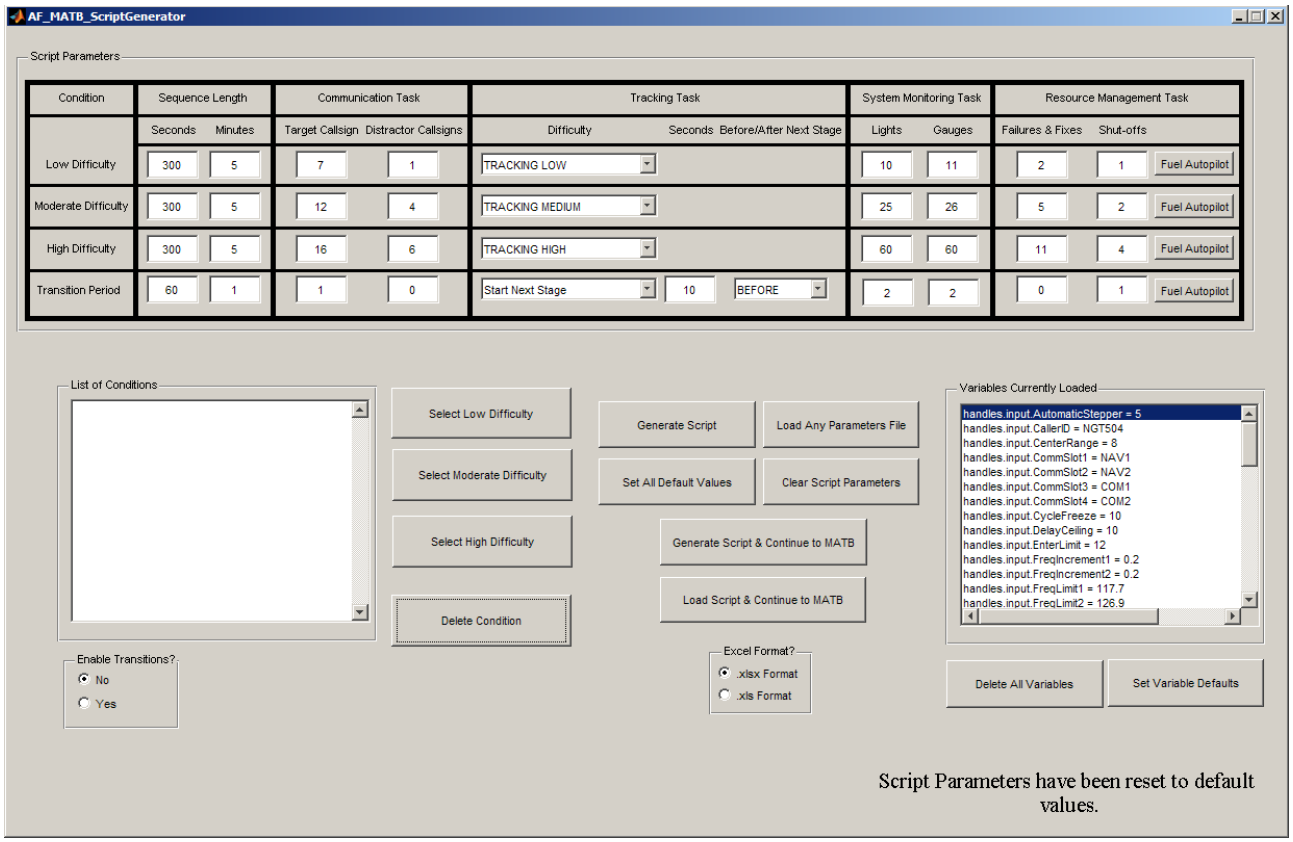


Figure 137 – The AF_MATB_ScriptGenerator window after the **Set All Default Values** button was pressed.

18.4. Clear Script Parameters

This button will clear the entry fields for all parameters in the GUI (see Figure 138).

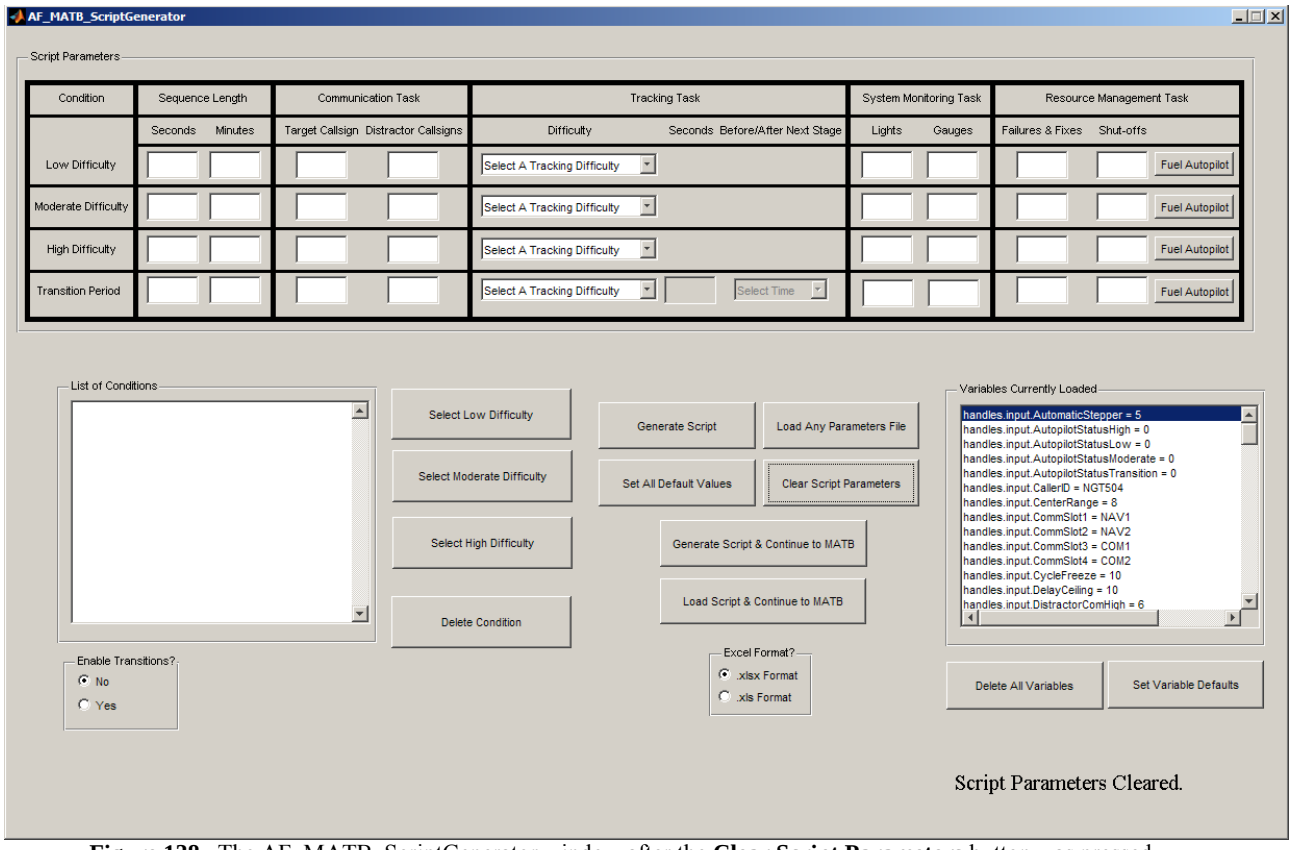


Figure 138 –The AF_MATB_ScriptGenerator window after the **Clear Script Parameters** button was pressed.

18.5. Generate Script & Continue to MATB

This button allows the user to save a script and immediately load that script into AF_MATB. This button is useful in situations where the experimenter is interested in testing a specific script.

To save a script, click **Generate Script & Continue to MATB**. Once all parameters have been verified, the user should select a directory and enter a filename, just as they would if the **Generate Script** button was pressed.

All of the rules that applied to the **Generate Script** button also apply here. If the script is constructed successfully, a message will be displayed in the GUI informing the user of that, followed by a message that AF_MATB is being loaded. Once AF_MATB has been loaded, the Script Generator GUI will dispose and AF_MATB will appear (see Figure 139).

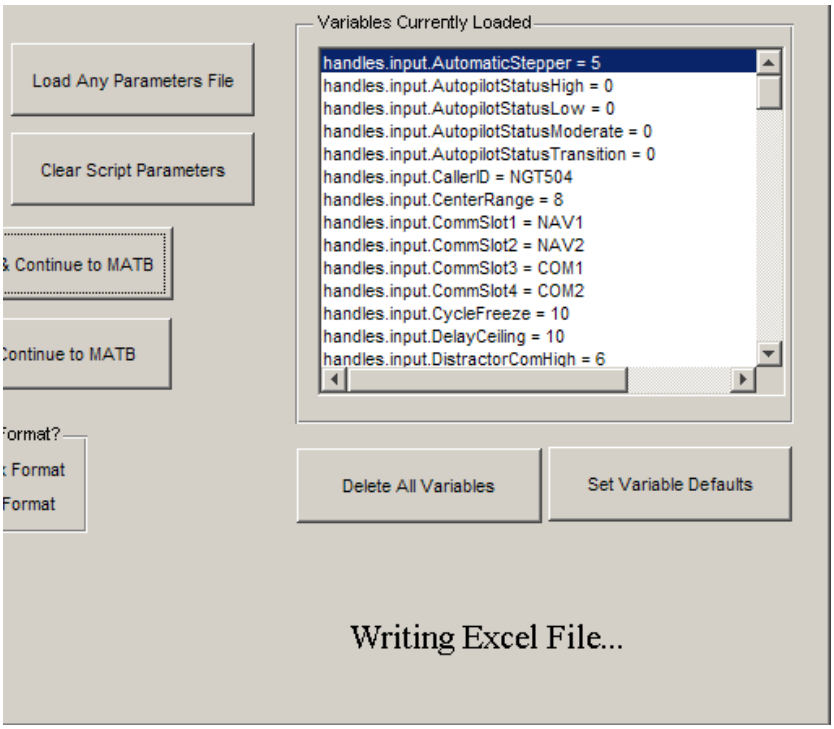


Figure 139 – When selecting **Generate Script & Continue to MATB**, the user will first be notified that the script is being constructed. Upon successful completion of script construction, the user will be informed that AF_MATB is loaded. Please see **Figure 140** for an example of that notification.

18.6. Load Script & Continue to MATB

This button will load any parameters file, as well as any successfully constructed script file. The parameters loaded from these files will be automatically updated in the appropriate fields, and then automatically loaded into AF_MATB. This button would be ideal for individuals who already have a saved script file and want to load it into AF_MATB immediately.

To load a file, click **Load and Continue to Script Generator**, select the directory and then the file using the file selection dialog; this process is the same as it would be if the **Load Parameters File** button was pressed.

All rules that apply to the **Load Parameters File** button apply here. If the script is loaded successfully, a message will be displayed in the GUI (see Figure 140). This message will inform the user that the script has been loaded and that AF_MATB is in the process of loading. Once AF_MATB has been loaded, the Script Generator GUI will dispose and AF_MATB will appear.

If the script does not load successfully, the script or task parameters file may be corrupt. These files may be unusable at this point, and it is recommended that you restart AF_MATB_ScriptGenerator.

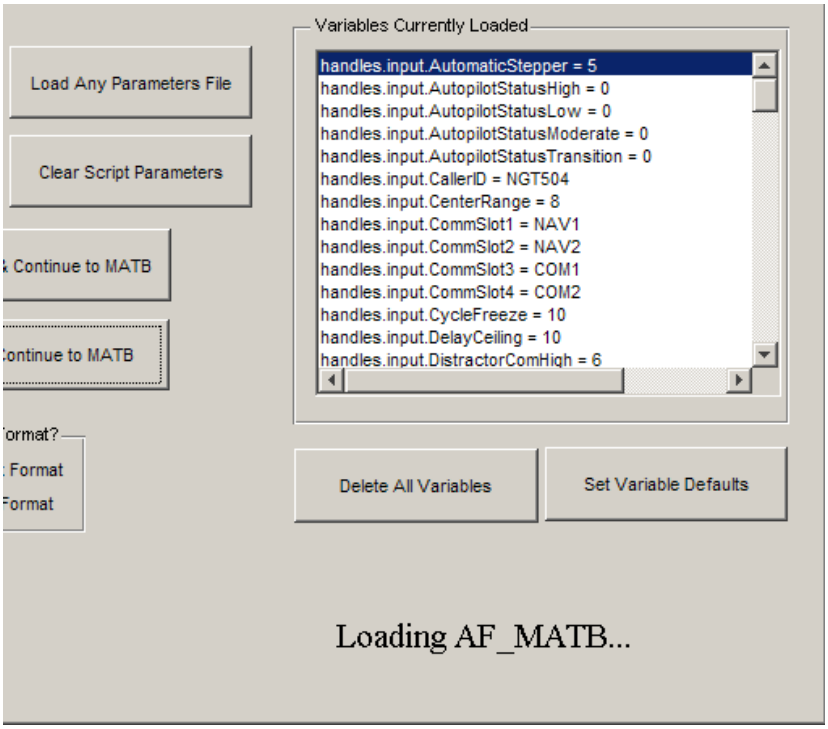


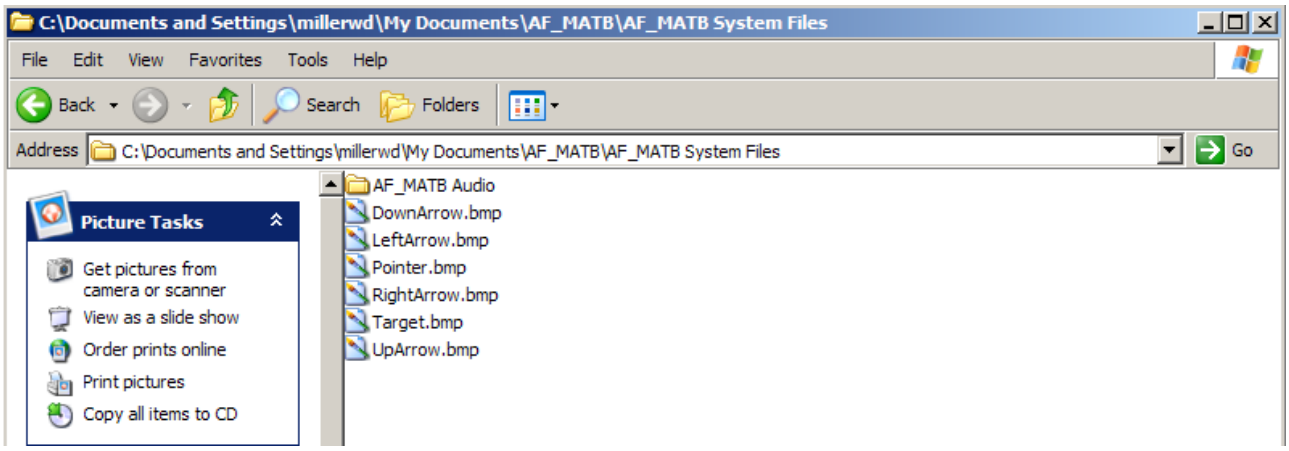
Figure 140 – Upon successful loading of a script file, the user will be notified that the AF_MATB task is loading.

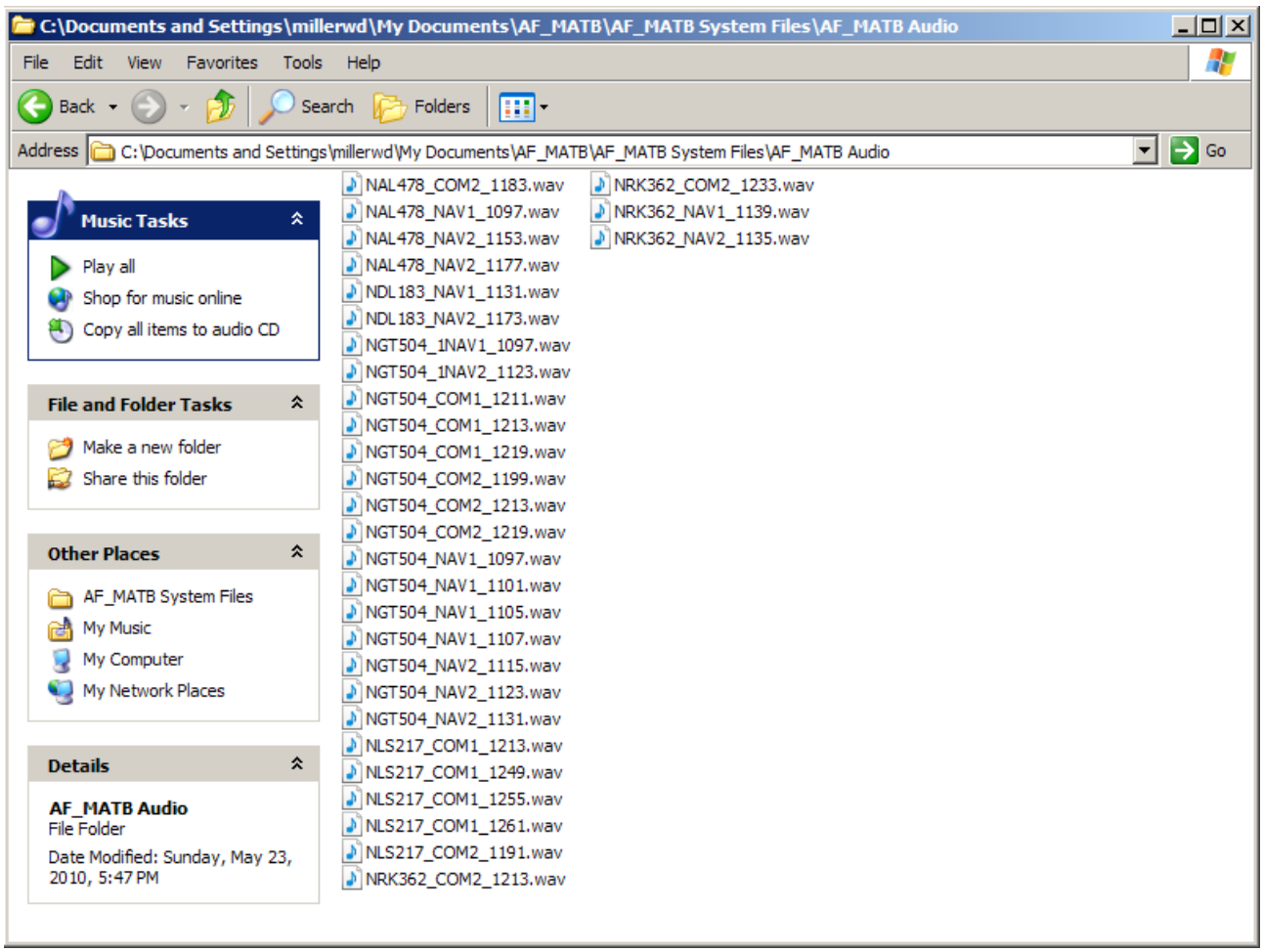
REFERENCES

Comstock, J.R., & Arnegard, R.J. (1992). The Multi-Attribute Task battery for human operator workload and strategic behavior research. NASA TM-104174, National Aeronautics and Space Administration, Langley Research Center, 1992.

APPENDICES

Appendix A: File Listing for AF_MATB System Files Folder





Appendix B: Example Performance File

Transition Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Event Time	Perceived or Actual Transition?	Previous Trial Difficulty	Current Trial Difficulty
300.059423	Actual New Trial	Low Difficulty	End of Experiment

Master Event Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Event Time	Task or Item	Action
13:54:05:328	User Start	AF_MATB Trial Started At This Time
0.533371	Tracking	Low Difficulty
2.948613	Pump 1	User Turned On
3.119595	Pump 2	User Turned On
3.363515	Pump 3	User Turned On
3.575366	Pump 4	User Turned On
3.783809	Pump 5	User Turned On
4.00709	Pump 6	User Turned On
27.498357	Communication	NGT504_NAV1_1107.wav
32.123613	Green (First) Light	Script-triggered Fault
34.732044	Green (First) Light	User Response
37.141955	Green (First) Light	Script-triggered Fault Timeout
38.389676	Pump 3	Script-triggered Fault
39.243324	Communication	Subject locked-in frequency 110.7 for Channel Nav 1
41.000049	Gauge 2	Script-triggered Fault
41.030948	Pump 1	User Turned Off
43.605659	Gauge 2	User Response
48.394632	Pump 3	Script-triggered Fault Timeout
49.922031	Red (Second) Light	Script-triggered Fault
50.923194	Gauge 2	Script-triggered Fault Timeout
51.259203	Red (Second) Light	User Response
54.89287	Red (Second) Light	Script-triggered Fault Timeout
55.460519	Communication	Subject locked-in frequency 118.1 for Channel Com 2

Entire log not shown due to length

Communication Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Event Time	Correct Channel	Correct Frequency	Target Signal	Timeout	Response Time	Response Channel	Response Frequency
27.704691	1	110.7	1	50.70469	39.210117	1	110.7
91.132662	1	109.7	1	114.1327	98.179835	1	109.7
112.290332	1	110.1	1	135.2903	121.959133	1	110.1
164.562737	2	112.3	1	187.5627	175.858754	2	112.3
205.702307	2	111.5	1	228.7023	215.595806	2	111.5
246.834648	1	109.7	1	269.8346	256.423618	1	109.5
275.573758	4	121.9	1	298.5738	289.498569	4	121.9

Correct Gauge Responses Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Event Time	Gauge Number	Response Time	Event Timeout
41.001239	2	43.610294	50.924478
68.030228	3	72.259872	78.01723
81.373351	1	84.71827	91.441224
127.50441	1	131.408399	137.522195
142.411499	2	146.863385	152.413654
158.788672	4	160.250142	168.791052
225.444538	2	226.939767	235.447104
227.074414	1	227.870041	237.10519
229.401312	3	235.013384	239.414146

Correct Light Responses Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Event Time	Light Number	Response Time	Event Timeout
32.126422	1	34.73512	37.143256
49.924901	2	51.262183	54.894152
80.029873	1	81.446823	85.05042
110.003857	2	111.638309	115.004323
156.850337	2	158.032261	161.870421
193.977708	1	196.683672	198.966434
195.020292	2	196.82735	200.040398
222.651464	1	223.799854	227.698091
234.902017	2	236.142986	239.906416

Fuel Values Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Fuel Target is: 2500

Event Time	Tank A Value	Tank B Value	Tank A Difference	Tank B Difference	Tank A Deviation	Tank B Deviation
1.203603	2483.003845	2483.003845	-16.996155	-16.996155	288.869269	288.869269
1.220703	2459.740135	2459.740135	-40.259865	-40.259865	1620.856764	1620.856764
1.236291	2459.221428	2459.221428	-40.778572	-40.778572	1662.89193	1662.89193
1.247778	2458.713188	2458.713188	-41.286812	-41.286812	1704.600811	1704.600811
1.25923	2458.329527	2458.329527	-41.670473	-41.670473	1736.428321	1736.428321
1.269705	2457.944515	2457.944515	-42.055485	-42.055485	1768.663794	1768.663794
1.283015	2457.600561	2457.600561	-42.399439	-42.399439	1797.71243	1797.71243
1.303	2457.092815	2457.092815	-42.907185	-42.907185	1841.026536	1841.026536
1.329508	2456.212992	2456.212992	-43.787008	-43.787008	1917.302097	1917.302097
1.347671	2455.543773	2455.543773	-44.456227	-44.456227	1976.356159	1976.356159
1.366093	2454.933667	2454.933667	-45.066333	-45.066333	2030.974363	2030.974363
1.387628	2454.32088	2454.32088	-45.67912	-45.67912	2086.582031	2086.582031
1.405428	2453.604895	2453.604895	-46.395105	-46.395105	2152.505782	2152.505782
1.423471	2453.015546	2453.015546	-46.984454	-46.984454	2207.538899	2207.538899
1.443301	2452.405506	2452.405506	-47.594494	-47.594494	2265.235867	2265.235867
1.462145	2451.745571	2451.745571	-48.254429	-48.254429	2328.489917	2328.489917
1.527456	2449.532431	2449.532431	-50.467569	-50.467569	2546.975494	2546.975494
1.594884	2447.072426	2447.072426	-52.927574	-52.927574	2801.328103	2801.328103
1.707252	2443.323868	2443.323868	-56.676132	-56.676132	3212.183994	3212.183994
1.816021	2439.723922	2439.723922	-60.276078	-60.276078	3633.205569	3633.205569
1.879094	2437.594862	2437.594862	-62.405138	-62.405138	3894.401259	3894.401259
2.002717	2433.475186	2433.475186	-66.524814	-66.524814	4425.550927	4425.550927

Entire log not shown due to length.

Incorrect Communication Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Response Time	Channel	Frequency
55.425971	4	118.1
95.69187	3	126.1

Incorrect Gauge Responses Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Response Time	Gauge Number
135.291842	2

Incorrect Light Responses Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Response Time	Light Number
65.964177	2

Tracking Coordinate Log

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consortium Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Center Coordinate is: 598 566

Event Time	X Coordinates	Y Coordinates	X-Axis Difference	Y-Axis Difference	X-Axis Deviation	Y-Axis Deviation
1.203628	601.633016	561.990045	3.633016	-4.009955	13.198807	16.079739
1.220718	604.811694	561.633593	6.811694	-4.366407	46.399176	19.065507
1.236307	607.990372	561.297792	9.990372	-4.702208	99.807529	22.110762
1.247794	611.16905	560.94134	13.16905	-5.05866	173.423868	25.590039
1.259246	614.347727	560.605539	16.347727	-5.394461	267.248193	29.100214
1.269722	617.526405	560.249087	19.526405	-5.750913	381.280502	33.073
1.28304	620.705083	559.892635	22.705083	-6.107365	515.520796	37.299903
1.303026	623.883761	559.556834	25.883761	-6.443166	669.969076	41.51439
1.329532	627.062439	559.200382	29.062439	-6.799618	844.625341	46.234802
1.347695	630.241116	558.843931	32.241116	-7.156069	1039.489591	51.209329
1.366118	633.419794	558.487479	35.419794	-7.512521	1254.561826	56.437972
1.387658	636.598472	558.131027	38.598472	-7.868973	1489.842046	61.92073
1.405452	639.77715	557.774576	41.77715	-8.225424	1745.330251	67.657604
1.423497	642.955828	557.418124	44.955828	-8.581876	2021.026442	73.648593
1.443327	646.134505	557.061673	48.134505	-8.938327	2316.930617	79.893698
1.462174	649.313183	556.705221	51.313183	-9.294779	2633.042778	86.392918
1.527482	652.491861	556.348769	54.491861	-9.651231	2969.362924	93.146254
1.594902	655.670539	555.992318	57.670539	-10.007682	3325.891055	100.153705
1.707273	658.849217	555.635866	60.849217	-10.364134	3702.627172	107.415272
1.81604	662.027894	555.279414	64.027894	-10.720586	4099.571273	114.930954
1.879114	665.206572	554.882298	67.206572	-11.117702	4516.72336	123.603295
2.002737	668.263256	553.89459	70.263256	-12.10541	4936.925139	146.540941

Entire log not shown due to length.

Performance Summary

AF_MATB V. 1.0
Programmed using MATLAB 7.5.0 and other Custom MATLAB Functions.

William Daniel Miller, Jr.
Air Force Research Laboratory
711th Human Performance Wing
Warfighter Effectiveness Directorate

Supported by funding from:
Consoritum Research Fellows Program
4214 King Street
Alexandria, VA 22302-1555

Performance Numbers:
System Monitoring

	Gauge 1	Gauge 2	Gauge 3	Gauge 4	All Gauges	Light 1	Light 2	All Lights	Total System
Event Occurences	3	3	3	2	11	5	5	10	21
Correct Responses	3	3	2	1	9	4	5	9	18
System Timeouts	0	0	1	1	2	1	0	1	3
System Errors	0	1	0	0	1	0	1	1	2
Correct Response Mean RT	2.68E+00	2.85E+00	4.92E+00	1.46E+00	3.10E+00	1.97E+00	1.44E+00	1.68E+00	2.39E+00
Correct Response STDev RT	1.66E+00	1.49E+00	9.78E-01	0	1.62E+00	8.02E-01	2.69E-01	5.96E-01	1.39E+00

Tracking

RMS Value	7.15E+01

Resource Management

Mean Deviation from Target Value	6.81E+02

Communication

Total Communications	8
True Communications	7
False Communications	1
Correct Responses	6
False Alarms	0
Response Timeouts	0
True Accuracy Errors	1
False Accuracy Errors	0
Unexplained Responses	1
No-Event Responses	1
Mean Correct Response Time	1.06E+01
StDev Correct Response Time	2.30E+00

Appendix C: Example Script File

Script Worksheet

Timeline (Seconds)	Script ID Code	Item Identifier	Event Trigger	Stop Time
0.05	8	0	32	0
0.05	345	3	49	0
27.43190039	12	12	63	50.4319
32.04384173	61	1	5	37.043842
38.29140929	97	3	9	48.291409
40.90784997	73	2	2	50.90785
49.87688147	62	2	6	54.876881
67.949407	72	3	3	77.949407
80.00261854	61	1	5	85.002619
80.70105105	12	10	76	0
81.34224476	71	1	1	91.342245
91.04191475	12	9	60	114.04191
109.9409005	62	2	6	114.9409
112.210767	12	10	61	135.21077
127.4572654	71	1	1	137.45727
133.0171476	72	3	3	143.01715
136.0093361	61	1	5	141.00934
142.3342401	73	2	2	152.33424
156.7837709	62	2	6	161.78377
158.7715572	74	4	4	168.77156
164.5093907	12	2	53	187.50939
193.891451	61	1	5	198.89145
194.9588776	62	2	6	199.95888
202.8987935	107	3	43	0
205.5953355	12	13	64	228.59534
222.6271581	61	1	5	227.62716
225.3872241	73	2	2	235.38722
227.0419907	71	1	1	237.04199
229.3757702	72	3	3	239.37577
232.5432083	95	1	7	242.54321
234.8027626	62	2	6	239.80276
237.9277803	74	4	4	247.92778
246.7722323	12	1	52	269.77223
275.5111512	12	8	59	298.51115

Event Translation Key Worksheet

Event Trigger Code	Event Description
1	Slider 1 Error
2	Slider 2 Error
3	Slider 3 Error
4	Slider 4 Error
5	Light 1 Error
6	Light 2 Error
7	Pump 1 Error
8	Pump 2 Error
9	Pump 3 Error
10	Pump 4 Error
11	Pump 5 Error
12	Pump 6 Error
13	Pump 7 Error
14	Pump 8 Error
15	Activates Autopilot Tracking
16	Fixes Slider 1 Error
17	Fixes Slider 2 Error
18	Fixes Slider 3 Error
19	Fixes Slider 4 Error
20	Fixes Light 1 Error
21	Fixes Light 2 Error
22	Fixes Pump 1 Error
23	Fixes Pump 2 Error
24	Fixes Pump 3 Error
25	Fixes Pump 4 Error
26	Fixes Pump 5 Error
27	Fixes Pump 6 Error
28	Fixes Pump 7 Error
29	Fixes Pump 8 Error
30	Activates Manual Tracking
31	Activates Autopilot Resource Management
32	Activates Manual Resource Management
33	Pump 1 Toggle for Resource Management Autopilot
34	Pump 2 Toggle for Resource Management Autopilot
35	Pump 3 Toggle for Resource Management Autopilot
36	Pump 4 Toggle for Resource Management Autopilot
37	Pump 5 Toggle for Resource Management Autopilot
38	Pump 6 Toggle for Resource Management Autopilot
39	Pump 7 Toggle for Resource Management Autopilot
40	Pump 8 Toggle for Resource Management Autopilot
41	Pump 1 Shut-off
42	Pump 2 Shut-off
43	Pump 3 Shut-off
44	Pump 4 Shut-off
45	Pump 5 Shut-off
46	Pump 6 Shut-off
47	Pump 7 Shut-off
48	Pump 8 Shut-off
49	Low Tracking Difficulty
50	Moderate Tracking Difficulty
51	High Tracking Difficulty
52	Correct Communication 1
53	Correct Communication 2
54	Correct Communication 3
55	Correct Communication 4
56	Correct Communication 5
57	Correct Communication 6
58	Correct Communication 7
59	Correct Communication 8
60	Correct Communication 9
61	Correct Communication 10
62	Correct Communication 11
63	Correct Communication 12
64	Correct Communication 13
65	Correct Communication 14
66	Correct Communication 15
67	Distractor Communication 1
68	Distractor Communication 2
69	Distractor Communication 3
70	Distractor Communication 4
71	Distractor Communication 5
72	Distractor Communication 6
73	Distractor Communication 7
74	Distractor Communication 8
75	Distractor Communication 9
76	Distractor Communication 10
77	Distractor Communication 11
78	Distractor Communication 12
79	Distractor Communication 13
80	Distractor Communication 14
81	Distractor Communication 15

Formatting changes were made to this example to allow it to fit the page.